



D20.5.1 (Final)

Analysis of a selection of educational ICT systems for mathematics

Main author : Jean-François Nicaud (UJF)

Nature of the deliverable : Report

Dissemination level : Public

Planned delivery date : October 2006

**No part of this document may be distributed outside the consortium / EC without
written permission from the project co-ordinator**

*Prepared for the European Commission, DG INFSO, under contract N°. IST 507838
as a deliverable from WP20
Submitted on 18-11-2006*

Summary

The deliverable describes a grid that has been designed to study ILEs (Interactive Learning Environments) in arithmetic and algebra, a process for choosing 6 ILEs to be studied and the result of the studies. The studied ILES are: ActiveMath, AnimalWatch, Cognitive Tutor, Math-Teacher Plus, MatXpert, Ms. Lindquist.

History

Filename	Status	Release	Changes	Uploaded
D20-05-01-F.pdf	Final	2	No change.	18/11/2006
D20-05-01-V1.pdf	Draft	1		31/10/2006

Contributor(s)

Name, First name	Contractor
Alshwaikh Jehad	(IOE)
Argyro Papadopoulou	(NKUA)
Cerulli Michele	(CNR)
Grugeon Brigitte	(UPARIS 7)
Maracci Mirko	(UNISI)
Trgalova Jana	(UJF)

Contents

1. Introduction	3
2. Design of a grid	4
3. A list of ILEs with brief descriptions	7
4. Choice of the ILEs to be studied	11
5. Results of the studies	12
ActiveMath.....	13
AnimalWatch	25
Cognitive Tutor	32
Math-Teacher Plus	47
MatXpert 1	57
MatXpert 2.....	65
Ms. Lindquist 1	71
Ms. Lindquist 2	80
6. Answer of the authors.....	88
7. Conclusion and future work	88
8. References	88

TELMA Teams

(ITD-CNR)	Consiglio Nazionale Ricerche – Istituto Tecnologie Didattiche – Genova - Italy
(UNILON)	University of London - Institute of Education – London - UK
(DIDIEM)	University Paris 7 Denis Diderot - DIDIEM – Paris - France
(NKUA-ETL)	National Kapodistrian University of Athens - Educational Technology Lab – Athens - Greece
(MeTAH)	MeTAH and Leibniz - IMAG, Grenoble, France
(Siena)	University of Siena - Department of Mathematics – Siena – Italy

1. Introduction

The deliverable presents a study of several didactical ICT systems in the field of arithmetic and algebra. The systems considered are interactive software. They will be named ILE (Interactive Learning Environment) in the rest of the deliverable.

The goal of this study is to have a good knowledge of the main existing ILEs for arithmetic and algebra, produced by research laboratories or companies, which are either products or advanced prototypes, are in use in several schools or have been experimented with a lot of students, and which have a high level of interactivity.

This study was conducted from January to October 2006 with three phases:

- 1) Design of a grid for studying the ILEs,
- 2) Production of a list of ILEs with brief descriptions,
- 3) Choice of 6 ILEs to be studied,
- 4) Study of the ILEs

The study consists of experimenting the ILE in the laboratory (without students), reading information about the ILE on the web, reading papers about the ILE. In principle, each ILE is studied by two persons of different teams. Sometimes, two independent grids have been filled. Sometimes, the second study has completed the grid of the first study.

The results of the studies have been sent to the authors of the ILEs to inform them and get feedback.

2. Design of a grid

The following grid has been established inspired by (Tchounikine 2004) and (Nicaud et al. 2006).

<i>Design and development</i>	
Name of the ILE:	
Date of the study:	
Who	
People/organisation involved in the design	
Interaction between these participants	
Why	
Aims of the design	
Research questions	
How	
Constraints affecting the design	
Underlying theories and principles	
Start of the design and number of versions	
Development status (advanced prototype, beta product, product)	
For whom	
Intended Audience	
Distribution	
License	
Price	

<i>Artefact</i>
Technical aspects <p>Operating System (or Web)</p> <p>Programming Language</p> <p>Translations</p>
Interface <p><i>Representations</i></p> <ul style="list-style-type: none"> * what rep. * nature (usual, new) * quality (poor, medium, rich) <p><i>Manipulations</i></p> <ul style="list-style-type: none"> * on what (objects, rules...) * quality (pleasant, painful)
Domain <p>Mathematical content</p> <p><i>Sort of exercises</i></p> <ul style="list-style-type: none"> * formal exercises * word problems
Types of activities <p>Lecture</p> <p>Examples solved by the ILE</p> <p>Exercises to be solved by the student</p>
Feedback <p>Immediate/delayed</p> <p>Nature</p> <p>Diagnosis</p>
Interactivity <p>Microworld</p> <p>Hypermedia</p>

Answer Multiple choice question Final open answer Step by step answer No answer	
Autonomy Degree of student's autonomy Self-evaluation	
Knowledge of the task The ILE knows what is the student's task The ILE can provide hint The ILE can provide the solution The ILE can solve the exercise step by step	
Teacher's role Parameterization Production of scenarios Production of exercises	
<i>Validation, usage, experimentation</i>	
Experimentation Usage analysis Qualitative results Quantitative results	
Usage Familiarization with the ILE	
Evolution	

3. A list of ILEs with brief descriptions

Below is the list of ILEs which was established from our knowledge of the field and by research on the Internet.

ActiveMath <http://www.activemath.org/>

Brief description: “ActiveMath is a stable, web-based, multi-lingual, user-adaptive, interactive learning system for mathematics. It is a Semantic Web application with a number of services. It employs technology for enhancing learning with scaffolding and instruction as well as with interactive and constructivist elements. The system provides an open architecture, knowledge representations, and techniques for the Web-presentation of interactive mathematics documents (hyperdocuments as well as printed material, slides as well as hyperbooks). Its modular architecture combines components such as a learner model, a course generator, a knowledge base and several integrated service systems. The course generator uses a formalization of the pedagogical knowledge for assembly of individual 'books' according to learner's goals, preferences and knowledge. ActiveMath's interactive exercise subsystem clearly separates components for evaluation and feedback. This adds flexibility, generality and transparency to exercises and supports action tracing and diagnosis. Special types of exercises and feedback are being developed to support the learner to overcome misconceptions and to improve her self-regulation and reflection.”

Algebrator <http://www.softmath.com/>

Algebraic problems solver.

Brief description: “Solves any problem from the algebra book. Helps learn algebra by encouraging the student to observe and ask for explanations. All the user does is click. At the same time the student finds out how algebra rules are applied in the given problem...”

Animalwatch <http://ccbit.cs.umass.edu/ckcold/researchpages/animalwatch.html>

Brief description: “Animalwatch is an Intelligent Tutoring System (ITS) that was developed to effectively teach pre-algebra arithmetic to elementary and middle school students. Animalwatch tutors addition, subtraction, multiplication and division of whole numbers, introduction to fraction concepts, addition of fractions, and subtraction of fractions. It maintains a student's proficiency estimate for each topic in the domain. Animalwatch is intelligent in that it provides students with problems for which the student is "ready". For example, a student is ready to see a word problem that involves fractions only after she has shown proficiency in all the topics that involve whole numbers. Also, there is a notion of difficulty of a problem within a topic that depends on other factors (size of the operators, number of steps involved in solving the problem, etc.). Animalwatch has been designed so that it is easy to add in new topics into the system. That is why it would be easy to extend it to algebra. It would require very little programming to extend the ITS to include problems of the type $2x + 3 = 11$ ”

Calcul numérique Seconde (CD-Rom)

Brief description: Exerciser: generates exercises in various domains (e.g. fractions, equations, factoring, inequalities, systems of 2 inequalities, ...) at various levels of difficulty (to be chosen by the user). Possibility to personalize the software by the teacher to adapt it to a pupil. Only in French.

Cognitive Tutor

<http://www.carnegielearning.com/products.cfm>

Description: “The research-based approach to improving student understanding of mathematical concepts. Straightforward design with no distractions, our easy-to-use interface gets straight to the mathematics. The Cognitive Tutor is based on research into how students learn and think. It uses real-world scenarios, so students understand how mathematics relates to their everyday lives. A variety of software tools engage students of all abilities and learning styles. The Tutor gives feedback and hints that help the student solve the problem, rather than just telling them “incorrect.” Skill bars show students what skills they have mastered, and where they need to improve, motivating the student to take responsibility for his or her own learning.”

GeoGebra <http://www.geogebra.at/>

Brief description: Dynamic geometry software with algebraic component enabling to solve analytic geometry problems geometrically, not algebraically (e.g. solving linear equations systems graphically).

LiveMath <http://www.livemath.com/>

Brief description: “LiveMath™ is a computer algebra and graphing system designed for the mathematician (or mathematics student or scientist) who is not necessarily a programmer.”

MathAid – Intermediate Algebra

<http://www.mathaid.com/>

Java-based package for e-learning.

Brief description: “MathAid teaching approach combines highly interactive and detailed tutorials with a system of self-tests. Every product includes two major parts: Basic Blocks and Self-Tests. Basic Blocks are designed to teach important concepts and methods for solving relevant problems. One Basic Block covers a particular topic, such as a theorem, a formula, or standard problems. Each Basic Block contains three types of lessons: Concepts, Interactive Examples, and Drills/Problems.”

When solving a problem, only the final answer is required, the user cannot solve the problem step by step.

Math Teacher + <http://www.mathkalusa.com/>

Brief description: “MATH-TEACHER helps students understand each concept before moving on to the next. Each lesson first presents the concepts and an infinite number of sample problems. MATH-TEACHER understands every step of a student's solution! Students type symbolic

expressions in a free-style format, MATH-TEACHER analyzes each line and indicates whether it is correct or not. This analysis detects common mistakes and provides focused help. The process repeats until the answer is reached. MATH-TEACHER perceives the proficiency of the student and generates problems accordingly.

Several levels of help are provided, including: Online textbook with a lesson on each concept, Sample problems with detailed solutions, General guidance on how to begin or continue a problem, Hints to solve a specific problem. Each module covers 1 to 1.5 years of mathematical concepts. The Customization Utility allows the instructor to fine tune and control the tutorial and on-screen tests. The timed On-screen Tests display all problems so the student can choose the one to answer next (duration, difficulty and scope can be customized). The Problems Generator for Printed Tests prints problems with and without answers, as homework or as hardcopy tests. The step-by-step answers are complete with explanations, charts, figures, graphs, and tables. The Calculator provided in MATH-TEACHER is designed to work with algebraic expressions. Rational expressions yield both rational and the approximate real results. Student scores are collected automatically to simplify record keeping.”

MathXpert <http://www.mathware.com/mathxpert.html>

Brief description : “MathXpert focuses on learning mathematics by helping the student to work through problems successfully.

Students often encounter difficulty in problem solving because they've made a trivial mistake that compounds in later stages, or because they've forgotten an important step of the strategy. Either of these problems result in the student becoming stuck, blocking momentum and halting the learning process. MathXpert is designed to eliminate these roadblocks by actively helping students solve any problem correctly.

The easy-to-use interface allows the student to focus on the correct strategy of problem-solving.”

Ms Lindquist: The Tutor <http://www.cs.cmu.edu/~neil/>

Brief description: “a new intelligent tutoring system for the domain of writing expressions for algebra "story" problems. This system is novel, because it is the first intelligent algebra tutor that combines a cognitive model of the domain with a rich pedagogical model of dialog-based tutoring. The algebra model is novel because contrary to prior work that has emphasized the difficulties of using variables, as well as the difficulties of comprehending the text of a word problem, I establish the empirical result that articulating a complete expression (e.g., $800-40*3$) is a major determinant of problem difficulty. The tutorial model is also novel because it is based on the observation of an experienced human tutor and captures the rich tutorial strategies specific to the domain of symbolization. The resulting system, called Ms. Lindquist, has been demonstrated to improve student learning.”

Study Works Mathematics Deluxe

<http://www.superkids.com/aweb/pages/reviews/math/7/swm2002/merge.shtml>

Brief description: “The program comes with practice problems and solutions that can be accessed on the computer and also includes a companion workbook that has problems and solutions in pre-algebra, algebra 1 and 2, geometry, pre-calculus and calculus. Installing the CD gives a

students access to lessons and problems, online tests and games, online homework help, an equation solver (to check homework for example), a graphing calculator, a unit convertor, software that can fit functions to large data sets, and an immense range of online tutorials in science and math.”

T-algebra

Brief description: “T-Algebra is an environment, enabling step-by-step problem solving in four areas: calculation of the values of numerical expressions; operations with fractions; linear equations, inequalities and equation systems; polynomials. The program supports the work of the student with the respective dialogue, diagnoses errors, and offers advice.

TREEFROG self-assessment software <http://java.cms.livjm.ac.uk/treefrog/treefrog.htm>

MS-DOS based program, not user friendly.

Brief description : “Rule-based system for students to practise their maths skills on. The PC-based system TREEFROG simply checks each step of the student's argument for consistency, and recognizes if the correct finishing point has been reached. The online prototype (which only covers polynomial algebra at present), gives rudimentary feedback using malrules if a mistake is made in a step.

A student is presented with an expression (say integral of $x \cos x \, dx$, or $x^2+3x+2 = 0$) and told to convert it to another form (function of $x + \text{constant}$, $x = ..$ or $x = ..$) which is equivalent. They can enter expressions one step at a time and at each step the system checks for syntax and logical errors.”

4. Choice of the ILEs to be studied

The criteria for choosing the ILEs have been:

- 1) Being either products or advanced prototypes,
- 2) Being in use in several schools or having been experimented with a lot of students,
- 3) Having a high level of interactivity,
- 4) Being available for the study,
- 5) Running in a language understandable by a member of the study team.

The following ILEs have been chosen:

ILE	1 st study	2 nd study
ActiveMath	Brigitte Grugeon	Argyro Papadopoulou
AnimalWatch	Jihad Alshwaikh	None (technical difficulty)
Cognitive Tutor	Jana Trgalova	Jihad Alshwaikh
Math Teacher +	Argyro Papadopoulou	Jana Trgalova
MathXpert	Michele Cerulli	Mirko Maracci
MrsLindquist	Mirko Maracci	Michele Cerulli

5. Results of the studies

This section is made of the grid filled in the study.

TELMA workpackage: ILE study

ActiveMath

A study of **Brigitte Grugeon-Allys**, DIDIREM-Paris 7, Paris, France
And **Argyro Papadopoulou**, ETL-NKUA, Athens, Greece

http://demo.activemath.org/ActiveMath2/main/viewBook.cmd?book=AC_UK_calculusCompleteRec&page=1

Design and development

Name of the ILE: ActiveMath

Date of the study:

21 September 2006

Who

People/organisation involved in the design :

The ActiveMath group was started in 2000 by Erica Melis (research in automatic proof planning), Jörg Siekmann (one of the founding father of AI in Germany) and Michael Kohlhase (the main developer of OMDoc, a knowledge representation for mathematical documents). Principal developers at that time were Paul Libbrecht, George Goguadze and Carsten Ullrich. In 2001, the group was extended among others by a software engineer (Stefan Winterstein).

Recently, Claus Zinn (previously at the Human Communication Research Centre at the University of Edinburgh) and Bruce McLaren (Carnegie Mellon University and Pittsburgh Science of Learning Center) joined the group. For a complete list, please see the members' page at <http://www.activemath.org/members.php>.

ActiveMath is mainly developed within the LeActiveMath project that involves eight partners ranging from didactical experts to computational linguistics (see www.leactivemath.org).

Most members of the group either work at the Competence Centre for e-Learning at the German Research Center for Artificial Intelligence or at Saarland University.

Interaction between these participants

They closely collaborate with pedagogical experts (within LeActiveMath, but also outside of our projects). Our development is not technology-driven, but motivated by and based on the real needs of the learners.

Researchers work in Activemath Group. E. Melis and her collaborators apply what they know to the assistance and improvement of self-directed and life-long learning, learning in schools and at universities with the vision to help learners, motivate them, and tap their full potential.

They apply what they know and learn in collaboration with psychologists, education practitioners, business and government enterprises and non-profit organisations to the assistance and improvement of learning in schools and at universities as well as life-long learning.

Why

Aims of the design

According to the authors, its aim is “to provide a web-based intelligent learning system for mathematics. Powered by semantical mathematics documents, based on founded pedagogical strategies, hoping to serve the self-responsible student by the free navigation and offering of mathematical and cognitive tools”.

Research questions

Technological highlights include (<http://www.leactivemath.org/index.php?id=6558>).

- open web-architecture and web-services
- reusable components
- multi-lingual system
- sophisticated personalization
- quality web-presentation including mathematical formulae
- player for several types of interactive (multi-step) exercises
- computer algebra system and other services used to generate feedback in interactive exercises
- interactive tools: lexicon, private and public notes, adaptive assessment tool, interactive concept map tool
- open learner model for visualizing the learner's abilities
- searchable web-based repository of interactive exercises.

Pedagogical highlights include:

- moderate constructivist course
- problem-based learning support
- to model pedagogical knowledge
- to model competency-level pedagogy.

Some examples of proposed thesis subjects (thesis to begin) :

- **Production and treatment of erroneous math-examples**
- **Exercise Generation with verbalization** : The generation of exercises in natural language will use templates for the mathematical content and enhance these by ontologies of several application domains (so that mathematically same exercises can be stated, e.g., for medical situations, technical situations, demographic situations etc). This is the basis for NL macro-planning and will be combined with a NL-micro planner that returns the actual natural language formulation of the exercise
- **Exercise evaluators and adaptive feedback**
- **Tutorial Component Mediator**
- **Agent-based evaluation and feedback mechanism**
- **Action Analysis in eLearning, Mining of Learner Logs**
- **Speech in ActiveMath** : In our tests with pupils we have encountered some children with deficiencies in reading. Teachers confirm this common and urgent problem. Affected pupils might be able to solve mathematical exercises, but they are somehow prevented from the information in a text exercise. Our goal is to provide a tool which reads exercises to a learner. There are some free speech tools at sourceforge. Your task is to examine, which one is suitable for ActiveMath and to integrate it. For the time being, reading normal text is enough. In a second step one might think about reading mathematical formulae and devise an implementation model

- **Added Value Mathematical Formulae Presentation :** ActiveMath presents mathematical content to a browser from a semantically encoded content. Among others, this allows the mathematical formulae to be presented with tool tips, click-to-explain-the-symbol, or copy-and-paste, all from the simple fact of encoding semantically the formulae (using OQMath, OMDoc, and OpenMath). We wish to extend this "value-added presentation of formulae" by considering supplementary input of the author. An example would be annotations about the variables in a well-defined context (e.g. "m is a the mass of the ball"), another avenue might be the usage of type-information about the symbols. The bachelor work should include the enrichment of a 10-15 pages "book" in order to convince mathematicians of the well-founded ness of the approach.

Remarks: It seems that proposed thesis subjects are not all worked. The research is narrowly lying to the evolution of the environment to improve interactions with student.

How

Constraints affecting the design

- First of all, *collaboration between experts from different fields*: through close collaboration with pedagogical experts and cognitive scientist, they ensure that the design is pedagogically sound and cognitively valid.
- *The principle of having a single system*: From a technological point of view, the constraints have been different depending on the research-project with the idea of a single system that provides an integrated learning platform enjoyable for the learner.
- *Implementation of a server*: to serve reasonable classrooms has occurred several times, peaks having been attained with 200 students. Each server was to run on a single computer and deliver an enjoyable experience (fast page download, fast advice, fast exercise evaluation, quality formula presentation, and coherent language).
- *Participative implementation*: Several experimental evaluations have been performed to assess the resulting platform. Currently, they are running a large scale study involving several hundred of learners in several European countries.

Underlying theories and principles

Moderate constructivism and competency-based pedagogy.

Remarks:

The authors have presented underlying theories and principles. There is staying at a level of general principal. It's difficult to see how the general principles are operationnalized.

Start of the design and number of versions

Things started in 2000, with a vision paper of Erica and Jörg.

Development followed with first alphas in 2002 until 2004 (about 5).

Since 2005 four betas have been delivered (last was in March of this year).

Development status (advanced prototype, beta product, product)

It's a product.

ActiveMath 1.0 should be released before November

For whom

Intended Audience : learners (and teachers), secondary school, university

Advanced high-school students (say, from 16 on) and University undergraduates. Earlier school orientation is being explored and experimented.

Distribution

The demos are globally accessible.

The server can be downloaded on request (currently per e-mail).

Installation instructions come along with installation times as short as 10 minutes. Installation is very easy.

License :

The betas have been distributed for academic public institutions under a license found in English translation at: http://www.activemath.org/InstitutsLizenz_en.html)

ActiveMath-1.0 and followers will be distributed under the German Free Software License: <http://d-fsl.de/>

Price

The product is not (yet) sold. Customization services are offered.

Artefact

Technical aspects

Operating System (or Web) : Web

ActiveMath server is a web-application and standalone tomcat server based on the Java 1.4 platform. Runs on most contemporary operating-systems. Clients are contemporary web-browsers with full support for Mozilla Firefox ≥ 1.5 and Internet Explorer > 6 and partial support for Safari ≥ 1.2 .

Programming Language :

java

Translations :

ActiveMath speaks, currently, English, German, Spanish, Mandarin, Russian, and Dutch. French is currently incomplete other languages have been reported to be added in about a day of work.

Interface

Representations

* what rep.

Textual, graphic, geometric, symbolic representations with hypertext links

*** nature (usual, new)**

Usual in maths, new because of interactive applets

*** quality (poor, medium, rich)**

High quality of symbolic and graphic representations

Manipulations

*** on what (objects, rules...)**

On graphics, on texts, numbers, formulas, matrices, geometric figures, applets for instance, for experimenting and conjecturing proprieties of notions (when parameters vary). It's possible to get semantically information from a given representation. There is also an editor for mathematical expressions. Hypertext is accessible at any time.

*** quality (pleasant, painful) : pleasant**

Domain

Mathematical content (according to the website presentation)

Differential Calculus

This book contains the complete content of differential calculus, on school levels as well as on university level. In addition, it contains all the required basics on functions and relations, on sequences and series, and on proofs (by induction). Numerous exercises, examples and interactive applets, including many from non-mathematical fields of application, support the comprehension of the mathematical contents.

School Mathematics: Fractions

Matheführerschein

Initiated by Fachhochschule Dortmund, the Matheführerschein content is meant for learners to prepare themselves for technical college/university. It incorporates about 300 exercises, grouped into "application", "problem solving" and "skills" areas, and an extensive library for self-exploration. Topics covered are: terms and equations, fractions, functions, differential and integral calculus. Matheführerschein employs a distinctive constructivist approach for learning, starting with complex real-world problems and leading to simpler yet related problems and needed skills on demand.

Westminster Calculus Course

The Westminster content is an implementation of the module Mathematics for Computer Scientists at the Harrow School of Computer Science. It includes interactive exercises and subject material for the module 2COS404. Topics covered are functions, graphs and matrices.

Statistics Analysis

A complete course in calculus, from real numbers definition to multivariate integration. Little set of MCQs and MuPad exercises.

Algebra

A brief introduction into groups.

Matheprisma

Combinatorics first part

Logic

This course comprises the basics of logic, such as logical operators, terms, formulas, sentences and laws. The main task of the course is to teach logic theory and its application in practice. According to the learner's field and interests, this course contains learning objects from biology, physics and mathematics.

Operation Research

Clear in its presentation, this course offers a comprehensive coverage of operations research along with real-world examples and problems. It is authored for economists, engineers, and mathematicians. It covers the following topics: Mathematical Modelling with Linear Programming, Network Planning, Game Theory, Dynamic Programming, and Queueing Theory and its applications.

Optimization Methods

This course considers theoretical, computational and applied aspects of optimization methods. It is authored for economists, engineers, and mathematicians. The course covers the following topics: Basic Concepts, One-Dimensional Search Methods, Multidimensional Search Methods, Methods of Constrained Minimization, and Integer Programming Methods. Examples of the specific problem solutions are given and illustrative interpretations of the obtained results are shown.

Metric Topology

A quick introduction to the topology of metric spaces

* Sort of exercises

- *Open question:* An open question allows to search “which is exclusively meant for a self-guide practise”. The open questions can be given with some hint or no.

An example: Find 5 different examples of *discontinuous functions* in economy!

- *Exercises with correction:* A correction with details is given about the exercise. Its' possible to select hypertext words to access definitions, theorems, other domains and also semantically information.

For example : We consider the following real function

$$f(x) = \begin{cases} \sin \frac{1}{x} & \text{if } x \neq 0 \\ 0 & \text{else} \end{cases}$$

Is f continuous at the point $x_0=0$?

- *QCM:* Several types of QCM are implemented
 - o QCM with several correct answers
 - o QCM with one correct answer

Hint is present.

- *Exercises (computational, identification, ..):*

For example: Computation of the slope of a straight line

The straight line $y=mx+b$ passes through the points $P=(1,8)$ and $Q=(6,3)$. Compute its slope m

There is a hint: “Use the formula for the slope pf a straight line”.

It's possible to access to formula with hyper text. It's possible to activate input Editor.

- *Erroneous exercises:*

An example: Eve wants to compute the derivative of the function $y = 1/(1-2 \cdot x)^2$ for $x \neq 1/2$. Her solution is $f(x) = (1 - 2 x)^3$. Her solution contains one or more errors. Please find the first error. Eve's solution: since $x \neq 1/2$ holds, the function is differentiable in its domain. She uses the Chain Rule for computing the derivative. The Chain Rule states that the derivative of a composite function $f \circ g$ can be calculated as follows $(f(g(x)))' = f'(g(x)) \cdot g'(x)$. Eve chooses $f=1/g^2$ and $g = 1 - 2 x$

Now, Eve calculates the first factor $(1/g^2)'$. She begins with rewriting $f = 1/g^2$ as $f = g^{-2}$ which leads to $f'(g) = (-2)g^3$

Then she calculates the second factor: $g(x) = -2$.

Finally, she combines the factors as follows:

$$= (-2) (1 - 2x)^3 (-2) = 4 (1 - 2x)^3.$$

Please find the error

In Active math there are many exercises like the above and there is also a report Design of Erroneous Examples for ActiveMath by E. Mellis in <http://www.activemath.org/pubs/Mellis-ErroneousExamples-AIED-2005.pdf>

*** formal exercises**

- Proof: Student has to organise the arguments of the proof. For solving it, student has to choose among several possibilities given by QCM.

Hint is present.

- Definition Exercise: Student has to choose a definition which is a answer to a question.

*** I word problems**

This is very small currently algebra so, we can't find formal exercises (factorisation, ...) and word problems

- There is no linear algebra except an optimization course (see the book at: <http://demo.activemath.org/ActiveMath2/main/viewBook.cmd?book=optimCustom&page=1>)

- There is a tiny introduction to monoids and groups (currently off line unfortunately) in English (an extract of Algebra Interactive!)- There is a tiny combinatorics course as well. The current flagship content is called "LeAM-calculus", and can be seen at [http://learn-](http://learn-review.activemath.org/ActiveMath2/main/viewBook.cmd?book=LeAM_calculusUniversityRec&page=1)

[review.activemath.org/ActiveMath2/main/viewBook.cmd?book=LeAM_calculusUniversityRec&page=1](http://learn-review.activemath.org/ActiveMath2/main/viewBook.cmd?book=LeAM_calculusUniversityRec&page=1). It actually contains a little amount of algebraic basics (e.g. relations and functions, binomial theorem).

Types of activities

Before looking at the types of activities proposed by ActiveMath, it is necessary to indicate that ActiveMath proposes two choices to work:

- with pre-recorded books
- or with personal book.

Lecture: There is an uniform and classical structure of the course including :

- Introductory notes motivating the study of the topic (for example, the tour de France to motivate the average slope to the derivative). The notes are very interesting and show that the authors would like getting mathematics accessible.
- A course with definitions, theorems and assertions, proofs, axioms, examples, exercises, notes and texts. Definitions, theorems, examples are often illustrated by graphic representations and by interactive applets. It's possible to experiment mathematics.

More, hypertext allows giving information (definition ...) at any time.

- Examples with many details to illustrate definition, theorem
- Proofs to demonstrate theorems
- Exercises (different types, solved or not)

It's possible to write personal comments on mathematics in personal books. There are accessible at any time.

Examples solved by the ILE

Several examples and solved exercises are given. They allow illustrating the study of topic. Java ActiveMath applet can be included according to the topics. Hypertext allows giving information at any time.

Exercises to be solved by the student :

There are several types of exercises (see below) and four difficulty levels which are indicated by stars ranging from * for very easy to **** for very difficult exercises.

- *Open question*, "which is exclusively meant for the self-guide practise. The open question can be given with some hint or no. When the exercise is over, the student can close the window and go on. It's possible to leave the exercise, at any time, even it's not finished.
- *QCM and exercises (computational, identification,)*:

Hint is often available.

Every answer is evaluated:

- If the answer is correct, the student can solve another exercise. The values of parameter are changed.
- If the answer is incorrect, another answer is asked. ActiveMath analyzes the answer and the principal errors are programming.

It's always to send a report about this exercise. No answer is given.

Feedback

Immediate/delayed

The student asks always for the evaluation of his answer. ActiveMath interface presents :

Evaluate

Input syntax help

Hint

The feedback depends of the context :

- Immediate evaluation of the answer to validate or invalidate it. Activemath tests if the student gives an answer.
 - If it's a correct proposition, ActiveMath recognizes it. According to the type of exercise, ActiveMath proposes another exercise in changing constants or another type of exercise. Always, ActiveMath proposes questions to test the student's opinion:

If you want, you can send a report about this exercise.

How do you feel about:

this exercise in general?	don't like it	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	like it
the effort you put in?	unsatisfied	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	satisfied
the result of this exercise?	disappointed	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	satisfied

- If it's incorrect :
 - If it's a classical error, ActiveMath recognizes it and proposes a note to explain it.
 - ActiveMath proposes a hint (one part of the strategy is given) : there are several levels, but the hint gives often a part of strategy or theorem to use, ..
 - After several steps, Activemath proposes to leave the exercise and to come back after. The correct answer is never given
- The hypertext is accessible at any time.

Nature

- Validation or invalidation
- Text to help student :
 - If answer incorrect help gives a part of strategy or theorem or formulas to use
 - If syntax error, an input syntax help gives information to the student to know how to input solutions
- Hypertext accessible at any time
- Interactive applet

Diagnosis :

ActiveMath is an **intelligent learning environment** that adapts to a mastery level: It tries to estimate **the current knowledge**, among others depending on the exercises the students solved, the texts the students read, etc. In the table of contents of a book, it's possible to see the estimation by **mastery colours** (Green yellow, red, black). The choice of exercises is done inside of a level.

ActiveMath produce a student "profile" but not like Pepite: the choice of exercises don't depend of calculated mathematic competences.

Depending on the type of book that the system generates for the learner, competencies are taken into account. For instance, for the book type "Train Competencies", the system can suggest to the learner what competencies to train (it makes this suggestion based on the past interactions), and assembles exercises that specifically train the selected competency. Additionally, in other books, the system always tries to select a variety of exercises, which train the whole range of competencies.

With "My Knowledge", it's possible to see how ActiveMath currently estimates a mastery of all available content.

Interactivity

Microworld

Some examples and solved exercises allow to experiment a mathematic notion (linear equation, quadratic function, ..) with interactive Applet

Hypermedia

Text has hypertext links to access definitions, theorems, other domains.

It's possible to get additional information from notes.

The system also allows extending personal books: the learner can let the system select exercises, examples, and text, adapted his current mastery level by using a menu attached to every concept. This selection uses the pedagogical knowledge represented in the system.

Answer

Multiple choice question :

See below

Final open answer

For all exercises.

Step by step answer

In case of incorrect answer for technical (computational, formal) exercises (see below)

No answer

Always possible

Autonomy

Degree of student's autonomy

Student's autonomy is high : selecting the topic of mathematics the student wants to learn more about, selecting the goal concepts of the course, choosing the hint, choosing when to change of activity or leave it. Finally, the student decides for the type of book.

Using Search it's possible to explore the complete mathematical content stored in ActiveMath. ActiveMath lets everybody search for simple texts, similar words, mathematical formulas or content items with specific features.

It's also possible to browse the relations between content items.

ActiveMath offers many other interesting features: ActiveMath allows to add notes to any content item, ActiveMath allows to create his own books tailored to his needs.

Self-evaluation

ActiveMath allows to self evaluate according to the level. ActiveMath asks for feelings.

Knowledge of the task

The ILE knows what is the student's task

Tasks and solutions have been implemented together with.

The ILE can provide hint

Yes, Activemath provides hint of different levels :

- Editing math expressions,
- Content (hypertext, annotations)
- the details of incorrect solutions and explanations,
- the details of steps of solution
- Questions for reformulation

The ILE can provide the solution

It can but it depends of the type of exercise and of the context.

The ILE can solve the exercise step by step

It's not proposed, but when the solution is incorrect, ActiveMath proposes a step by step decomposition of the task, from the hint

Teacher's role

There is no teacher module.

Parameterization

No

Production of scenarios

No

Production of exercises

No

Validation, usage, experimentation

Validation

Usage analysis

ActiveMath is constantly being evaluated and tested under various conditions. It has been already evaluated in classrooms a number of times. In particular, it was used for half a year at the Bellevue School in Saarbrücken. There, it was employed during school to teach the mathematics of fractions.

Usability studies have been performed in 2002, 2004, and 2005 with lab experiments. Real usage has been done in 2003 and 2005.

Qualitative results

Usability studies lead to user-interface design recommendations. Major issues of the user-interface being too rich for school pupils also lead to adaptations especially for it.

Quantitative results

I know there are more such results but they have not been really published unfortunately.

The only somewhat formal ones I know of are of the formative lab-evaluation in 2005 where, overall, was rated enjoyable (80%), would like to use again (90%), and better than other maths software (80%).

Usage

Familiarization with the ILE :

It's enough easy.

Evolution

Two principles propositions

- to allow learning in school at long term,
- To go on research in the group and outside

Application projects after LeActiveMath are proposed.

TELMA workpackage: ILE study

AnimalWatch

A study of **Jehad A. O. Alshwaikh**, University of London, United Kingdom

Design and development

Name of the ILE: AnimalWatch

Date of the study: 2001

Note: the homepage for this software (<http://ccbit.cs.umass.edu/AnimalWatch>) doesn't work so that many of the needed information are not valid.

Who

People/organisation involved in the design

- The University of Massachusetts at Amherst and The Center for Knowledge Communication (<http://ckc.cs.umass.edu/ckc/>). The National Science Foundation supported this project.
(<http://ccbit.cs.umass.edu/ckcold/researchpages/animalwatch.html>)
- Professor Carole Beal and Computer Science Research and Associate Professor Beverly Woolf (http://www.cs.umass.edu/csinfo/newsletter/images/sigbits_spring03.pdf, p. 5)

Interaction between these participants

From the information written on the leaflet with the CD and from the website, it is clear that many participants involved producing this software. The main important thing I noticed is the field tested for the software with students in three schools.

Why

Aims of the design

- “to raise young girls’ self-confidence in their math skills”
- Presents basic mathematics and fractions in word problem.

Research questions

Women performance and engagement in mathematics, science and technology.

How

Constraints affecting the design

Underlying theories and principles

- Eccles, J. S., Wigfield, A., Harold, R. D., & Blumenfeld, P. (1993). Age and gender differences in children's self and task perceptions during elementary school. *Child Development*, 64, 830-847:
"students' motivation and in turn their achievement in a particular academic subject such as math is thought to reflect their attitudes about the domain, including their liking of the subject, their belief that it is important and valuable to learn, and their concept of their own ability to master the subject."
- Integration between mathematics, science (Biology) and technology.
- Learning through exploration and collaboration.
- ZPD (Vygotsky)

Start of the design and number of versions

"The tutor has been tested in numerous trials between 1996 and the present" (<http://ckc.cs.umass.edu/ckc/>)

Development status (advanced prototype, beta product, product)

As mentioned in the "changes" file in the CD, 9 versions have been developed.

For whom

Intended Audience

9-12 years old. Actually there are more ages suggested through my search in the related websites. The main website suggests that this software has been designed for the young girls' self-confidence and was tested with grades 4-6 (Arroyo et al., 2003). Other related studies stated that the Animalwatch is for 8-11 years old (Arroyo, Schapira & Woolf, 2001).

Distribution

License

Copyright 2001 University of Massachusetts at Amherst, and the Center for Knowledge Communication.

There is a license by the Sun Microsystems, Inc.

Price

In the NSF report it is written that the AnimalWatch cd-rom "is available by request or can be downloaded from the website, but the download time may be prohibitive for most users."

Artefact

Technical aspects

Operating System (or Web)

PC: windows 95 or later

Mac: MacOS 8.0 or later

Programming Language

??? (Java(tm) Runtime Environment, Version JRE 1.1.8)

Translations

Interface

Representations

*** what rep.**

There are five representations:

1. words (natural language) give information (information box) relates to the problem/question
2. pictures or images box also relates (or don't) to the problem/question,
3. Graphs/charts/diagrams that relate to the problem and the students need to look at and understand them in order to solve the problem.
4. symbolic (numerals) to provide the problem
5. The answer box(s): empty box(s) for student to fill the answer with numbers,
6. Animated diagrams in the help screen to help students in understanding the solutions of addition (units/tens) and fractions.

*** nature (usual, new)**

There are some new representations especially the information box.

*** quality (poor, medium, rich)**

Medium quality in general. But there are good representations especially in the help screen such as the units/tens or fractions since they are animated and diverse.

Manipulations

*** on what (objects, rules...)**

- units/tens: base-10 manipulative to add numbers
- illustrations of algorithms (for subtracting and multiplication procedures)
- Fractions: illustrations in addition and subtraction.
- diagrams

*** quality (pleasant, painful)**

It closes to be pleasant more than painful! The main problem is that the manipulation solves the problem totally with **minimum** interaction with the students when it asks the students to enter a specific number in blank boxes and if the students make mistakes again it solves the problem and the students **must** enter the right answer to proceed.

Domain

Mathematical content

- Basic arithmetic in integers: addition, subtraction, multiplication, and division
- Fractions: identification of the numerator, denominator, or both (the whole fraction. Addition and subtraction of fractions.
- Word problems

Sort of exercises

* formal exercises

NA

* word problems

all of the problems are word problem based on given information about three endangered species (Whale, Panda, and Horse)

Types of activities

Lecture

NA, but all problems are word problems based on given information about three endangered species

Examples solved by the ILE

When students make mistakes the AnimalWatch suggests three clues and offers later the solution of the problem.

Exercises to be solved by the student

The entire ILE problems are supposed to be solved by the students.

Feedback

Immediate/delayed

Immediate feedback upon the students enters the answer in the box answer. There is also a delayed feedback that the teacher gives for the students in her/his progressive after finishing the session.

Nature

The student gets either “correct” or hints to solve the problem after answering the problem, and later, if s/he still doing wrong the help screen will solve the problem.

Diagnosis

ILE enables teachers to monitor their students’ progress by “Animal Watch Reporter” included within the CD. Only teachers can give the feedback to students (i.e. students can’t reach to the Reporter by themselves). One essential problem is that the progress report doesn’t offer the previous assessment for the students and just display the last assessment upon the students’ work in other words; the students are only evaluated on their final answer entered in the answer box.

Interactivity

Microworld

Students interact with ILE through reading the information, the help and the problems; and by writing the answer in the answer box.

Hypermedia

There are two ways of presenting the information: text and pictures. I would say that this ILE is not hypermedia since there are no sounds, videos, or www interactive

Answer

Multiple choice question

NA

Final open answer

NA

Step by step answer

The help screen presents the answer step by step but the students' involvement in the solution process is very limited and, unfortunately, the students are only evaluated on their final answer entered in the answer box

No answer

ILE presents the answer always.

Autonomy

Degree of student's autonomy

Very limited. The students can choose only the beginning of the software (which animal only) and later the tutor guides the students. However, the students might work by themselves and without teachers' help.

Self-evaluation

It is feedback more than self-evaluation.

Knowledge of the task

The ILE knows what is the student's task

The ILE itself offers the tasks/problems and is adaptable for teachers to create their own tasks for the students but this adaptability is reduced given the lack of availability of the website.

The ILE can provide hint

When the students answers wrongly, this feedback is given in two hints : the first is general and gives the clue about the needed arithmetic operation (difference, ..) and the second is very specific and direct (the difference between 7 and 1, ..)

The ILE can provide the solution

If the student keeps solving the problem wrongly, the software offers an indirect solution at the early stage but later the ILE solve the problem directly.

The ILE can solve the exercise step by step

If a student keeps solving the problem wrongly and after 3 trials, the AnimalWatch solves the problem step by step.

Teacher's role

Parameterization

Production of scenarios

Production of exercises

As written in the "readme" file, teachers can produce their exercises.

Validation, usage, experimentation

Experimentation

As indicated in the website of The University of Massachusetts at Amherst and The Center for Knowledge Communication (<http://ckc.cs.umass.edu/ckc/>), many research and papers were inducted and produced within the production process of AnimalWatch (see the reference for examples.)

Usage analysis

Qualitative results

Quantitative results

Usage

Familiarization with the ILE

It is easy to cope with this ILE and to be familiar with it and this is a plus for the AnimalWatch software.

Evolution

"The tutor has been tested in numerous trials between 1996 and the present" (<http://ckc.cs.umass.edu/ckc/>), and as mentioned in the "changes" file in the CD, 9 versions have been developed but, unfortunately, there are no information about these kinds of changes during these developments.

Reference:

I. Arroyo, A. Schapira and B. Woolf. " Arroyo, I., Beck, J., Beal, C. & Woolf, B. (2003). Learning within the ZPD with the AnimalWatch intelligent tutoring system.

Arroyo, I., Schapira, A., & Woolf, B. (2001). [Authoring and Sharing Word Problems with AWE \(Animalwatch Web-based Environment\)](#). In J.D. Moore, C.L. Redfield and W.L. Johnson (Eds.), [Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future](#), pp. 527-529. Amsterdam: IOS Press.

Beal, C. & Arroyo, I. (2002). The AnimalWatch project: Providing adaptive teaching for students and adaptive support for teachers.

<http://odtl.dcu.ie/mirror/crite/CRITEfinal.html>

<http://reports-archive.adm.cs.cmu.edu/anon/hcii/CMU-HCII-05-104.pdf>

TELMA workpackage: ILE study

Cognitive Tutor

A study of **Jana Trgalova & Marie-Caroline Croset**, Leibniz Laboratory, Grenoble, France
And **Jehad A. O. Alshwaikh**, University of London, United Kingdom

Design and development

Name of the ILE: Cognitive Tutor®

Date of the study: First study: June 2006, second study: October 2006

Web site: [1]

Who

People/organisation involved in the design:

Carnegie Learning™, The Cognitive Tutor Company (Pittsburgh, USA)

Research group Advanced Computer Tutoring Project, Carnegie Mellon University, Pittsburgh (cognitive psychology, computer science)

Practicing classroom teachers

Interaction between these participants:

- According to Corbett (2002), the research team cooperated with practicing teachers throughout the process of developing, piloting, evaluating, and disseminating the tutor.
- As a result of designing the tutors, a 'company called Carnegie Learning, Inc. has been formed to market the Cognitive Tutors and curricula' (Aleven & Koedinger, 2002)

Why

Aims of the design: Improve student achievement in mathematics

Initially :

- Learn more about skill acquisition –rather than to produce practical classroom results.

Then :

- Optimize learning
- Reference to a Cognitive Model
- Shape procedural knowledge acquisition
- Reflects the National Council of Teachers of Mathematics curriculum

More specially, for Algebra II :

- Support students in applying algebra to real-world problems
- Support reasoning among multiple representations
- Employ modern computational tools

Research questions: Contribute to the understanding of human cognition.

Initial motivation:

- Test the ACT* theory: see whether it is possible to produce successful learning by getting students to act like the underlying production-rule model.

Later:

- How to develop tutors which will implement an externally specified curriculum, which can be deployed in a wide range of classrooms and which leave students with a competence that makes a demonstrable contribution to their activities outside of the specific domains taught by the tutor.

How

Constraints affecting the design

Core commitment: instruction should be designed with reference to a cognitive model of the competence that the student is being asked to learn.

To avoid combinatorial explosion: requirement that each student action be on an interpretable path and restriction on possible interpretations of the behavior. In case of ambiguity about the interpretation of the student action the student was presented with a disambiguation menu to identify the proper interpretation of the action.

Later, the requirement that the student stays on an interpretable path has been relaxed (Anderson & al., 1995).

Underlying theories and principles:

- Cognitive psychology: the nature of human learning and performance, the ACT theory of skill acquisition.
- Computer science (research on artificial intelligence techniques)
- Cognitive tutors embody the framework reported by the National Research Council report on educational assessment (NRC, 2001), that three essential pillars support scientific assessment: a general model of student cognition, tasks in which to observe student behaviour, and a method for drawing inferences about student knowledge from students' behaviours. (Corbett, 2002)

ACT* Theory

Strong basis in research into how people learn, remember, and solve problems : *ACT* theory* (Anderson, 1983), and later *ACT-R theory* (Anderson, 1993) of learning, memory and performance. ACT is a general theory of cognition that focuses on the acquisition of cognitive skills which consist in units of goal-related knowledge. The main principles of this theory are:

1. *Procedural and declarative knowledge*: cognitive skill depends on converting goal-independent declarative knowledge (that can be acquired by being told) into goal-related production rules which represent the procedural knowledge (that can only be acquired by doing).
2. *Knowledge compilation*: production rules can only be learned by employing declarative knowledge in the context of a problem-solving activity.
3. *Strengthening*: both declarative and procedural knowledge acquire strength with practice.

8 principles for the design of computer tutors:

1. Represent student competence as a production set (accurate cognitive model of the target skill).

2. Communicate the goal structure underlying the problem solving (instructional objective: exposing and communicating the set of goals and sub-goals to the student)
3. Provide instruction in the problem solving context (context specificity of learning)
4. Promote an abstract understanding of the problem-solving knowledge (correct abstraction reinforced in the language of help and error messages).
5. Minimize working memory load (by minimizing presentation and processing of information not relevant to the target production; by designing a curriculum in a way a few new things are taught at a time)
6. Provide immediate feedback on errors
7. Adjust the grain size of instruction with learning
8. Facilitate successive approximations to the target skill

NCTM recommendations:

Cognitive tutors embody the framework reported by the National Research Council report on educational assessment (NRC, 2001), that three essential pillars support scientific assessment: a general model of student cognition, tasks in which to observe student behaviour, and a method for drawing inferences about student knowledge from students' behaviours (Corbett, 2002).

Greater emphasis is put on:

- problem solving
- reasoning among different mathematical representations
- communication of mathematical results

Start of the design and number of versions

1983: LISP and Geometry tutor

mid 1980s: the beginnings of building Algebra Tutor

late 1980s: the evaluation process

1990s: the development of Algebra tutor

Number of versions:

Development status (advanced prototype, beta product, product)

Products: Algebra I and II, Geometry, Integrated Maths I, II and III

For whom

Intended Audience:

Middle school and high school students in general and students who need help in classrooms in particular. The tutors are also offered to the college students.

Distribution

License

Carnegie Learning curricula can be purchased as a full license for software and text or as a software-only license. Maintenance & support as well as training apply to all purchases [2].

Price (from Carnegie Learning web site [3])

For each of the 5 curriculum products (see Domain) :

Student Text Set \$21,25

Teacher Text Set \$80,00

Software Implementation Guide \$47,50

Complete Curriculum Kits \$99,00

Artefact

Technical aspects

Operating System (or Web):

Windows 2000, XP; Mac OS 10.3

Broadband internet connexion is required

Programming Language

Translations

N/A

Interface

Note: The analysis below concerns Algebra II tutor, in particular Systems of Equations.

Representations

* Natural language, table (numerical representation), graphic and algebraic

Example: Systems of equations - Section 2

A problem is given in English (*verbal or natural language representation*, see appendix). The text describes the problem situation and a few questions are asked. There is also a suggestion to define a variable for a given quantity and to write expressions in terms of this variable.

Students answer the questions by filing in the worksheet (*table representation*, see appendix). They identify relevant quantities in the problem situation (in this situation, TIME, NUMBER OF DOLLS ASSEMBLED, and DEMAND), label the columns accordingly, enter appropriate units in the second row of the worksheet (Remark: sometimes, the appropriate unit is entered automatically, e.g., when the student enters the labels “DOLLS ASSEMBLED” and “DEMAND”, the unit DOLLS is completed automatically), enter a symbolic formula for each quantity in the third row (*algebraic representation*), and answer the questions in the successive table rows. The algebraic expressions can be entered at any time during the problem solving process and the student does not need to rely on them to answer the questions. The individual questions are intended to help scaffold the algebraic modelling early in the curriculum. Later, the student should use these expressions as problem solving tool and enter them early in the problem. The emphasis is on using the expressions both in symbol manipulation and to automatically generate values in the worksheet (Corbett & al., 2000). E.g., to answer the third question “*At the current rate, in how long will 75000 dolls have been sold?*”, the student can set up the equation $17615 + 555T = 75000$, solve it in the Solver window and type the answer, 11477/110, into the question 3 cell of the TIME column, where it will be converted to 104.3364. Or s/he can type directly the arithmetic expression $(75000 - 17615) / 550$ where it will be computed automatically. Once the value of T has been typed in this cell, the worksheet automatically generates the corresponding number of dolls assembled, 61084.0909.

To answer some of the questions, the student may need to solve an equation. S/he can do this in the Solver window (*algebraic representation*, see appendix). S/he sets up the equation and solves it by performing a series of transformations to be chosen from a Solver list. Then s/he types the result into the appropriate worksheet cell.

Students graph the functions corresponding to the expressions with the graphing tool (*graphical representation*, see appendix). They label the x- and y-axes (in this situation, TIME and DOLLS respectively), indicate what the line will represent by labelling the symbol of the line (in this

situation, one line will represent the DEMAND and the other the number of DOOLS ASSEMBLED). The grapher generates the lines automatically. It is not necessary to enter algebraic expressions before asking for drawing the lines.

*** nature (usual, new)**

All representations are quite usual. The issue of transfer was in mind in the interface design. Two points have been considered: (1) the skill the students will learn with the tutor is problem solving in a particular interface, and, (2) the interface students learn will have a large impact on where their skills will transfer. Nowadays, there is an increasing tendency for the target skill to involve use of computer software. Thus, part of the competence taught in the current algebra tutor is how to use spreadsheets (Worksheet), symbol manipulation packages (Solver) and graphing routines (Grapher).

*** quality (poor, medium, rich)**

Point of view of the authors of the first study: rich (comments below):

- multiple representations available at the same time: helps making connexions among them
- independent from each other: the student can start solving the problem from either point of view
- no possibility to navigate among the windows (e.g., it is impossible to copy an expression in the Worksheet window and paste it in the Solver window)
- graphing tool works as a black box: the student controls only the boundaries for the values of variables

Point of view of the authors of the second study:

In one hand, I think that the quality of these representations is medium in the sense of the choices being offered for the students in order to solve the problem. For instance, in addition to the scenario (the problem), there are the glossary, the grapher and the solver windows. These representations offer good opportunities for the students to experience more than one kind of representing the problem.

However, in the other hand, the quality of these representations is poor, since it does not pay a significant attention to the students' language or communication issues. They also present maths within the stereotype, i.e. its language is numbers, diagrams and symbols.

Manipulations

*** on what (objects, rules...)**

Worksheet

- the student must fill in the table with labels and units of the quantities involved in the problem situation and with algebraic expressions that model the situation
- to answer the questions, he may use either numerical expressions that are computed automatically or algebraic expressions. If the unknown is the output value, once the input value entered in the appropriate cell, the corresponding output value is generated automatically.

Solver

- allows solving equations by applying transformations rules to be chosen from a given list: the student chooses a rule (e.g., divide both sides...) and enters the argument (in this case, the number both sides of the equation will be divided by).

Grapher

- the student enters labels for the coordinate axes and for the lines to be represented, sets up

ranges for the variables and labels the point of intersection

*** quality (pleasant, painful)**

Comparing to other tutors (AnimalWatch as an example), these tutors offer many representations. These representations also offer chances for the students to be involved in solving the problem by entering the answer, trying to draw the equation, multiply or add by the calculator, or find the meaning of a concept or a shape from the glossary.

Domain

Mathematical content:

Bridge to Algebra (Number Sense and Algebraic Thinking, Fractions and Mixed Numbers, Decimals, Ratio and Proportion, Percents, Integers, Algebraic Problem Solving, Geometric Figures and Their Properties, Area and the Pythagorean Theorem, Probability and Statistics, Volume and Surface Area, Linear Functions, Number Systems, Transformations)

Algebra I (Organizing Single Variable Data, Simplifying Linear Expressions, Finding Linear Equations from Graphs, Solving Linear Equations and Inequalities, Standard Form, Slope Intercept Form, Mathematical Modeling, Linear Expressions and Equations, Quadratic Expressions and Equations, Solving Systems of Linear Equations Algebraically and Graphically, Solving and Graphing Equations Involving Absolute Values, Problem Solving using Proportional Reasoning, Analyzing Data and Making Predictions, Powers and Exponents)

Geometry (Area and Perimeter, The Pythagorean Theorem, The Geometry of Right Triangles, Triangles, Parallel and Perpendicular Lines, Similarity and Congruency, Quadrilaterals, Circles, Surface Area and Volume, Geometric Transformations and Symmetry, Right Triangle Trigonometry)

Algebra II (Linear Functions, Systems and Inequalities using Modeling and Multiple Representations, Matrices, Modeling and Multiple Representations - Quadratic Expressions and Equations, Cubic Expressions and Equations, Exponential Functions, Quadratic Functions, Higher Order Polynomials and Rational Functions, Exponentials, Sequences and Series, Complex Numbers, Periodic Functions, Functional Notation, Probability, Conic Sections)

Integrated Math (Statistics : Inferential and Descriptive Statistics, Measures of Central Tendency and Dispersion, Modeling Data Using Regression Equations ; Probability : Simple and Compound Probabilities : Theoretical and Experimental Probabilities, Sample Spaces, Venn Diagrams, Permutations and Combinations ; Algebra : Linear Functions, Quadratic Functions, Exponential Functions, Higher Order of Polynomial and Rational Functions, Sequences and Series, Trigonometric Functions, Parametric Functions ; Geometry : Area and Perimeter, Geometry and Algebra of Right Triangles, Transformation and Symmetry, Similarity, Parallel and Perpendicular Lines ; Discrete Math ; Numbers and Operations)

Sort of exercises

*** formal exercises**

Examples:

Graph the equations $-5x + 6y = 50$ and $x + 4y = 42$, then find their intersection.

Simplify the expression $(4x^2y^4)(3x^3y^3)$.

Rewrite the expression $x^2 + 7x + 10$ as the product of two factors.

*** word problems**

Example: see appendix

Types of activities

Software-based individualized computer lessons with collaborative real-world problem solving activities (3 days/week of classroom instruction, 2 days/week teacher-supervised software lab environment)

Examples solved by the ILE

There are demos.

Exercises to be solved by the student

Both formal exercises and word problems are provided to be solved by the students.

Feedback

The feedback is immediate and immediate error correction is required.

Nature

- **Help message on demand.** The student can ask for help at any step. Help messages are generated from templates associated with the correct productions which would have fired at that point. There are usually 3 levels of help provided (see below).
- **Error message** (red font flag) without any explanation: if the student makes a general mistake, the feedback is given in red colour; therefore s/he knows that there is something wrong.
Example: In the above situation, when one enters an incorrect expression that the tutor cannot interpret, the expression is highlighted in red, but no message is given. However, the student can ask for help and he gets progressively the following messages: “*Enter an expression to calculate the number of dolls assembled using the time.*” / “*First consider the initial value of the number of dolls assembled. Next consider how the number of dolls assembled will change for each day.*” / “*Write an expression that means the same thing as the initial value of the number of dolls assembled plus the change in the number of dolls assembled for each day times the time.*” / “*Write an expression that means the same thing as 35000 plus 250 times the time.*” / “*Enter $35000 + 250X$.*”
- **Explanatory error message** (Just-in time help) if the mistake matches a common misconception. There is no attempt to provide any deep diagnosis of the cognitive origins of the error, but rather explain why it is an error. Example: in the situation of the above example, if one tries to type “DOLLS” as the one of the quantities, the label is displayed in red font and one gets the message “*Be more specific. Can you say more about DOLLS?*”

3 levels of feedback on demand:

- (1) **A hint on a goal to be accomplished.** In the above example, if the student asks for a hint at the beginning of the problem solving process, s/he gets the message: “One quantity in the problem statement is described in terms of another quantity. Which quantity is stated directly in the problem statement?”
- (2) **A general advice on achieving the goal.** In the example above, the following message is: “Label this column (the column is being highlighted) by clicking on the box for the label and typing in a short phrase describing a quantity in the problem.”
- (3) **A concrete advice on how to solve the goal in the current context.** In the same example, the following (and the last) message is: “Try typing in the phrase “the time””.

Diagnosis

The ‘*Just-in-time*’ *help/feedback* might be considered as a diagnosis tool. In this feedback, as mentioned above, the tutor helps the students to solve the problem instead of telling the students that their solution is ‘incorrect’.

Model Tracing – The underlying cognitive model is employed to interpret each student action and follow the student's individual solution path through the problem space, providing just the support necessary to complete the problem successfully. At each step the student action is compared to applicable rules in the model and immediate feedback is conventionally provided. If the student action matches an applicable rule, the tutor accepts the action and fires the rule to update the internal representation of the problem state. If the student action does not match the action of any applicable rule in the model, the action does not register and the tutor provides a brief message in the hint window. The student can ask for advice at any step. The tutor provides 3 levels of advice: (1) an advice on the goal to be accomplished, (2) a general advice on achieving the goal, and (3) a concrete advice on how to solve the goal in the current context.

Knowledge tracing – Bayesian Network: The cognitive model is also employed to monitor the student's growing knowledge in problem solving. Knowledge tracing assumes a two-state learning model. Each coding rule is either in the learned state or in the unlearned state. Performance in applying a rule is governed by its learning state, but only probabilistically. At each opportunity to apply a rule in problem solving, the estimate of the probability for the rule is updated, contingent on whether the student action was correct or not. These probability estimates are displayed in the skill meter on the screen.

The **skillmeter** window will indicate the progress of the student's skills.

Interactivity

After choosing the mathematical subject to work with, the students are offered a problem to work on. This problem is offered in the scenario window. The students are supposed to read and try to answer the problem in the worksheet window. If the students need to calculate, they might use the solver window. The students also might check at the glossary window any mathematical concept they do not understand from the problem. After finishing the problem, the diagnosis window (the skillmeter) will indicate the progress of the student.

Answer

The problem has to be solved within each representation. For example, in the above situation, the student must fill in all the cells of the Worksheet table and draw lines and label the point of intersection in the Grapher window.

Step by step answer

Yes, by providing hints in 'On-demand help'

No answer

The final answer is provided by the tutor at the end of the 'On-demand help' if the student asks for it.

Autonomy

Degree of student's autonomy

Point of view of the authors of the first study:

The software is not meant to be used autonomously, without teacher's intervention. The student can need help from the teacher, for example to interpret some hint messages. However, due to the feedback provided, the student can work alone with the software.

Point of view of the authors of the second study:

Even though one of the main aims of the tutor is to offer individual learning where the students might work by themselves and without teachers' help, I think that this autonomy is limited in the sense of 'controlling one's learning'. The students can choose only which subject to learn in specific areas of maths and later the tutor guides the students. For example, there are no 'back or forward' buttons. However, the 'glossary' and the 'solver' windows are very helpful towards this autonomy.

Self-evaluation

The tutor provides the student with evaluation in two different way :

- Immediate feedback about the student's answer (see 'feedback' above),
- Skills development by 'skillmeter' windows where the degree of 'mastering' a specific rule or skill is represented by the progress in the bar (see appendix).

Knowledge of the task

The ILE knows what is the student's task

Information in working memory is of two kinds: (1) information about what the current state of the problem is, and (2) a representation of what the goal is.

The ILE can provide hint

Yes, see feedback.

The ILE can provide the solution

Yes, by providing hints and help (see feedback).

The ILE can solve the exercise step by step

Yes, by 'on demand help' (see feedback).

Teacher's role

Note. The version of Cognitive Tutor that is being analysed does not allow any parameterization, nor production of exercises or scenarios. There exists a Teacher's Toolkit, an application for performing administrative tasks associated with the Cognitive Tutor programs. Such tasks include:

- Creating, maintaining and deleting class rosters
- Tracking students progress as they work through the curriculum
- Restarting and skipping problems
- Changing a student's placement within the curriculum [3]

For college students, the tutors are programmed to be individual learning or unsupervised. While in schools environment, the teachers' role is to encourage students and to help the students while they supervise them. The idea is not to give the answer but to encourage the students to ask the teachers rather than to copy the answer offered by the tutor (Corbett, McLaughlin & Scarpinato, 2000). Moreover, the cognitive tutors are designed that the students will work at their own on the software which constitutes 40% of the instruction time.

Parameterization

N/A

Production of scenarios

N/A

Production of exercises

N/A

Note: There is a special website for the teachers involved in the cognitive tutors curriculum (<http://k12.carnegielearning.com/>). Through this website, teachers have the access to the resources, the cognitive tutors educational materials and the discussion forums. This website requires username and password.

Validation, usage, experimentation

Experimentation

Many experiments have been done since the first trial of the tutors. One of the main experiments was conducted in the 2001-2002 school year when the Cognitive Tutor Algebra and Geometry courses were used at about 700 sites and by more than 125,000 students in 38 states. This includes urban, suburban, and rural middle and high schools, both public and private (Corbett, 2002). Experiments were aiming at evaluating the tutor lessons from 3 points of view:

- Learning rate
- Validity of the underlying student model
- Pre-test to post-test learning gains

Usage analysis – a few results

Geometry Tutor:

- 1985-1986 school year: pilot study
large achievement gains
- 1986-1987 school year: major test comparing classes with the tutor with classes without the tutor but with the same teacher
positive results when students could work one per machine
- 1993: preliminary evaluation of a geometry tutor completed based on the cognitive model of geometry proof
large positive results but only for the teacher integrated into the project

Algebra Tutor:

- 1987-1988 school year:
no differences between experimental classes with the tutor and control classes without
- 1989: evaluation of a word algebra tutor
some large positive results in the laboratory
- since 1995: a newer word algebra tutor is used in Pittsburgh Public Schools
significant achievement gains

Quantitative results

- According to the website of the cognitive tutors [5], the students using the Cognitive Tutor Algebra I program:
 - Have a 70% greater likelihood of completing subsequent Geometry and Algebra II courses
 - Achieve 15-25% better scores on the SAT and Iowa Algebra Aptitude Test
 - Experience equivalent results for both minority and non-minority students
- ‘Cognitive Tutors have been shown to be 1 standard deviation better than classroom instruction (Anderson et al., 1995; Koedinger et al., 1997) and have been estimated to be 1.75 standard deviations better than self-study (Corbett, 2001).’ (Aleven & Koedinger 2002, p. 150)
- According to Koedinger et al. (2000), in comparison to students in traditional mathematics classes:
 - Students taking Cognitive Tutor Algebra I have been shown to perform 85% better on average on assessments of complex mathematical problem solving and thinking.
 - Students taking Cognitive Tutor Algebra I have been shown to perform 14% better on average on standardized assessments of basic mathematical skills.
 - Students completing the three-course Cognitive Tutor sequence (Algebra I, Geometry and Algebra II) have been shown to perform better on the TIMSS assessment (by 30%) and real-world problem solving assessments (by an average of 227%).
 - Students who enrolled in Cognitive Tutor Algebra I have been shown to be 69% more likely to pass traditional Geometry and 71% more likely to pass traditional Algebra II.
 - The benefits of the Cognitive Tutor Algebra I approach have been shown to be equivalent for both Caucasian and African-American students.
- According to Corbett (2002):
 - ‘College students working with APT completed a problem set three times faster and scored 25 percent higher on tests than students completing the same problems in a conventional programming environment.’
 - ‘High school students in geometry classes that employed GPT for in-class problem solving scored about a letter grade higher on a subsequent test than students in other geometry classes who engaged in conventional classroom problem-solving activities.’
 - ‘High school students in the Cognitive Tutor Algebra I course scored about 100 percent higher on tests of algebra problem solving and reasoning among multiple representations, and about 15 percent higher on standardized assessments than similar students in traditional Algebra I classes.’

Usage

Familiarization with the ILE

Evolution

1983: Beginning of the work for Lisp tutor and Geometry Tutor

1984: First experimental courses with the two tutors

1985: Integration of geometry tutor in a real classroom

Mid 1980s: development of Algebra I tutor

2001-2002 school year: Algebra and Geometry tutors in use at about 700 sites by more than 125000 students in 38 states

Bibliography

Aleven, V., Koedinger, K. (2002) An effective metacognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science*, 26, 147–179

Anderson, J. R., Corbett, A. T., Koedinger, K. R., Pelletier, R. (1995) Cognitive tutors: Lessons learned, *The Journal of Learning Sciences*, 4, 167-207.

Corbett A. (2002) : Cognitive Tutor Algebra I : Adaptive Student Modeling in Widespread Classroom Use. In *Proceedings from a Workshop Technology and Assessment : Thinking Ahead*.

Corbett A., McLaughlin, M., Scarpinato, K. C. (2000) Modeling Student Knowledge: Cognitive Tutors in High School and College, *User Modeling and User-Adapted Interaction* 10, 81-108.

Koedinger, K., Anderson, J. R., Hadley, W. H., Mark, M. A. (1997) Intelligent Tutoring Goes To School in the Big City, *International Journal of Artificial Intelligence in Education*, 8, 30-43.

Koedinger, K., Corbett, A., Ritter, S. & Shapiro, L. J. (2000). *Carnegie Learning's Cognitive Tutor™: Summary Research Results*. Available at: <http://www.carnegielearning.com/>, 25/09/06.

Web sites

[1] <http://www.carnegielearning.com>

[2] http://www.carnegielearning.com/company_quickquote.cfm

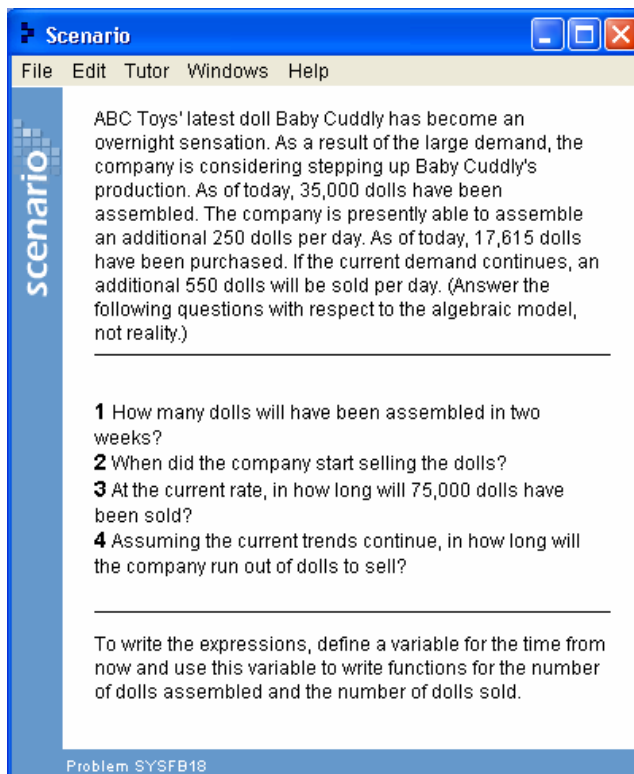
[3] http://www.carnegielearning.com/services_textbook_form.cfm

[4] http://www.carnegielearning.com/research/research_reports/

[5] <http://www.carnegielearning.com/approach.cfm>

Appendix. Algebra II Tutor (Systems of Equations / Section 2)

Scenario window (natural language representation)



Scenario

File Edit Tutor Windows Help

scenario

ABC Toys' latest doll Baby Cuddly has become an overnight sensation. As a result of the large demand, the company is considering stepping up Baby Cuddly's production. As of today, 35,000 dolls have been assembled. The company is presently able to assemble an additional 250 dolls per day. As of today, 17,615 dolls have been purchased. If the current demand continues, an additional 550 dolls will be sold per day. (Answer the following questions with respect to the algebraic model, not reality.)

- 1 How many dolls will have been assembled in two weeks?
- 2 When did the company start selling the dolls?
- 3 At the current rate, in how long will 75,000 dolls have been sold?
- 4 Assuming the current trends continue, in how long will the company run out of dolls to sell?

To write the expressions, define a variable for the time from now and use this variable to write functions for the number of dolls assembled and the number of dolls sold.

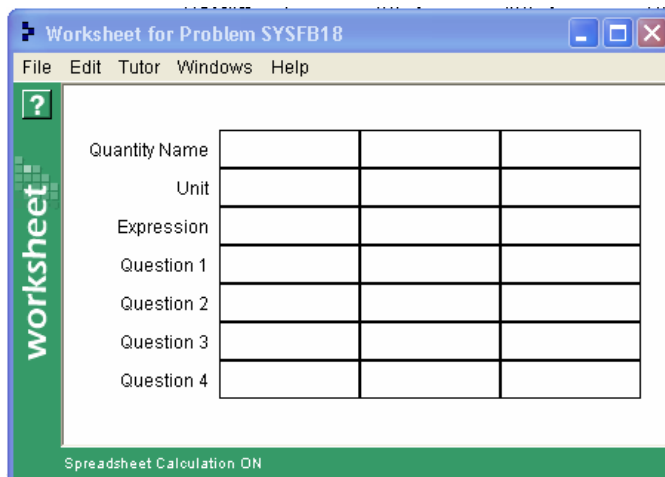
Problem SYSFB18

Description of the problem situation

Questions

Suggestion to choose a variable and write expressions using this variable

Worksheet window (table and algebraic representations)



Worksheet for Problem SYSFB18

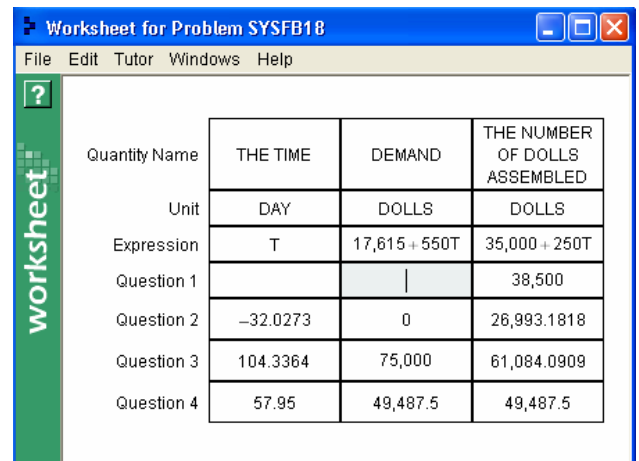
File Edit Tutor Windows Help

worksheet

Quantity Name			
Unit			
Expression			
Question 1			
Question 2			
Question 3			
Question 4			

Spreadsheet Calculation ON

To solve the problem, the student fills in the worksheet. S/he is required to label a column before filling in any cell in the column. There are no other constraints on the order in which the cells are filled in.



Worksheet for Problem SYSFB18

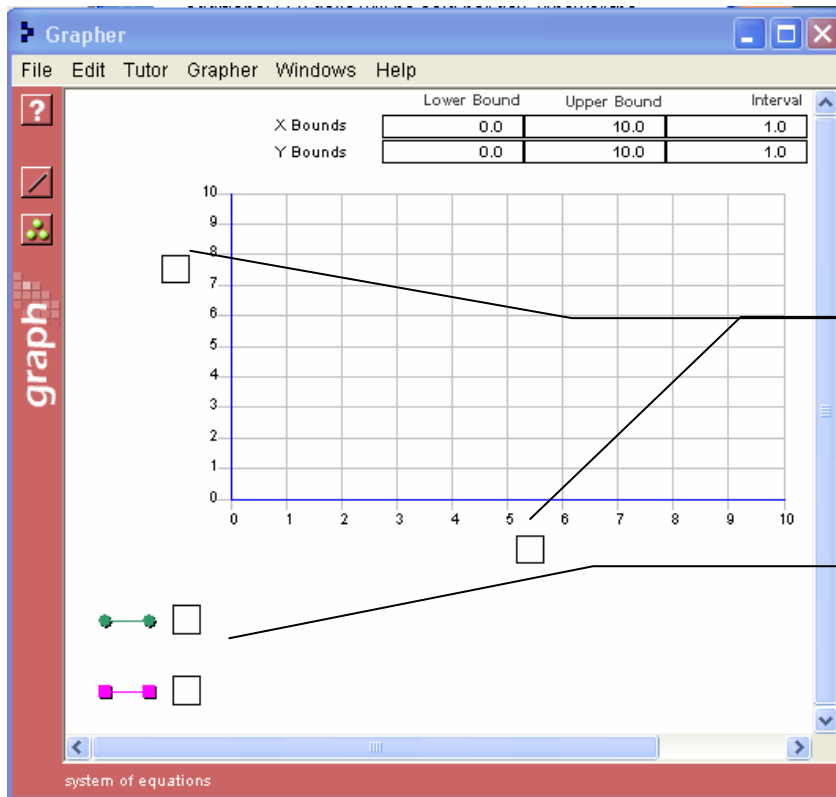
File Edit Tutor Windows Help

worksheet

Quantity Name	THE TIME	DEMAND	THE NUMBER OF DOLLS ASSEMBLED
Unit	DAY	DOLLS	DOLLS
Expression	T	$17,615 + 550T$	$35,000 + 250T$
Question 1			38,500
Question 2	-32.0273	0	26,993.1818
Question 3	104.3364	75,000	61,084.0909
Question 4	57.95	49,487.5	49,487.5

The worksheet near the conclusion of the problem.

Grapher window (graphical representation)

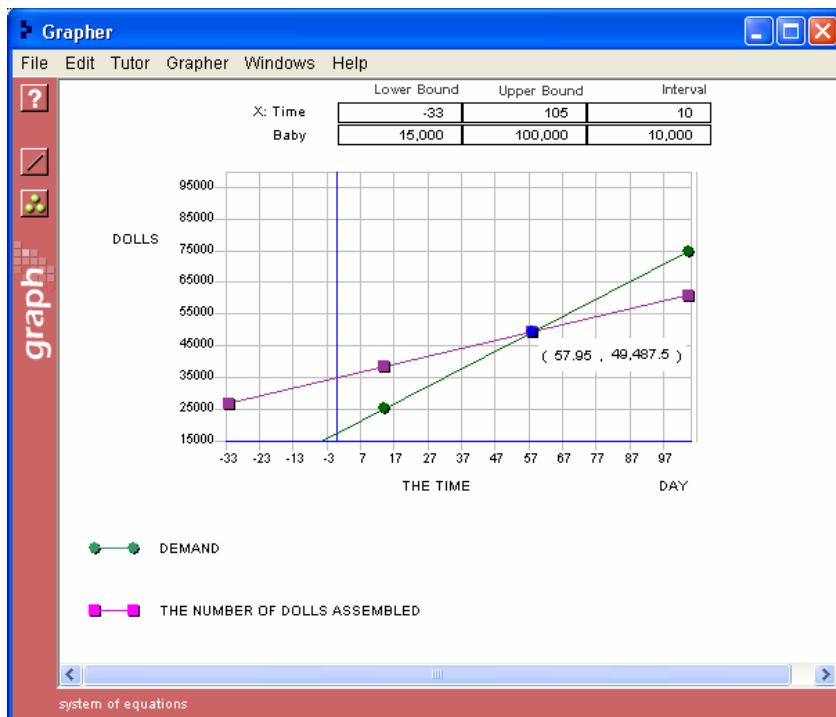


The student sets up appropriate values for x and y variables ranges.

The student labels the x- and y-axes.

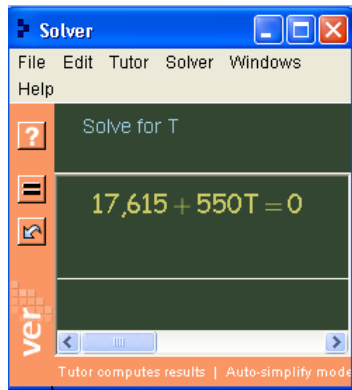
The student enters the labels of functions to be represented.

Grapher window with the solution of the problem

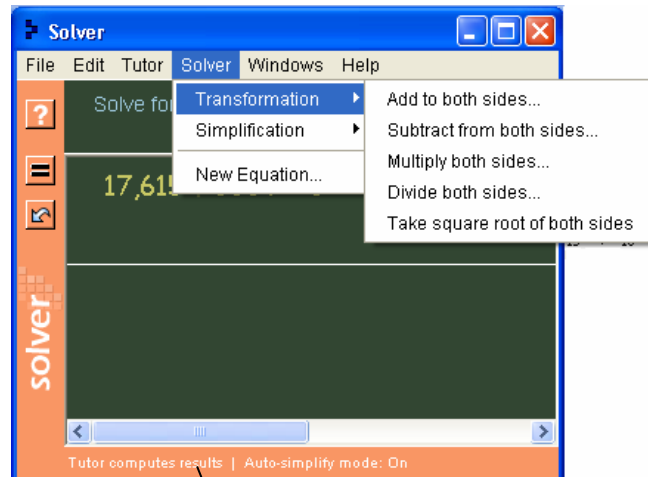


The lines are drawn by the Tutor. The student can display a few points on both lines and write the coordinates of the intersection point.

Solver window

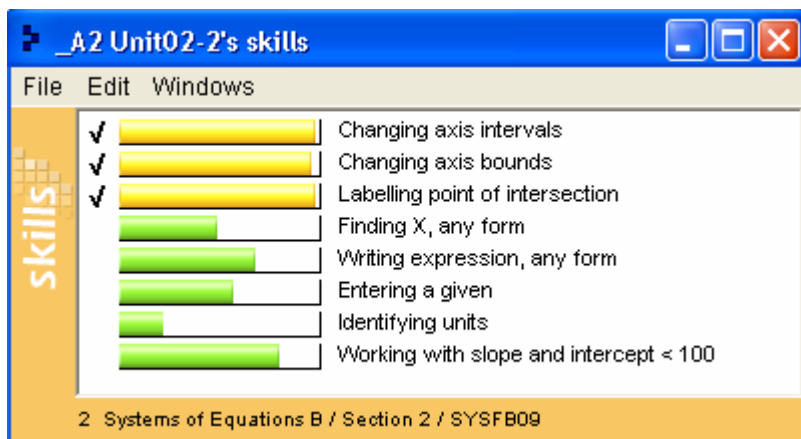


The student enters the equation to solve ...



... and solves it by applying the transformations to choose from a given list.

Skill meter window



Skills to be learnt in the activity. The first three skills are supposed acquired by the student.

TELMA workpackage: ILE study

MATH-TEACHER PLUS

A study of **Argyro Papadopoulou**, ETL-NKUA, Athens, Greece
And **Jana Trgalova**, MeTAH-IMAG, Grenoble, France

Design and development

Name of the ILE: MATH-TEACHER PLUS (<http://www.mathkalusa.com/index.html>)

Date of the study: Start of the study June 1 - Final version August 22

Second study: September 2006 (study based on the demo version)

Who

People/organisation involved in the design: Publisher Hickerson Education, Inc., MathKal Educational Software Ltd.

Interaction between these participants: In 2004, Hickerson Education, Inc. purchased MATH-TEACHER PLUS from MathKal Educational Software Ltd. Hickerson, Inc. now owns all title, rights, and interest in MATH-TEACHER PLUS.

Why

Aims of the design: to create an interactive tool that allows students to master skills and algebraic concepts and understand the ideas behind the concepts taught in the classroom.

(Brief description: The mastering skills process is interactive. A problem is presented to the student who must provide an answer, which is not necessarily the final one. The program leads the student to the final answer by providing guidance, hints and comments on each step of the solving process. In addition, the program includes lessons, accompanied by many examples in each of the topics.)

The primary purpose of MATH-TEACHER is for additional practice in topical areas as students learn the concepts in their regular program. There are tutorial explanations in each program. [Note: the publisher says that these modules can serve as complete courses, but I think it would be too difficult for most students to use them in that way.]

Research questions: Universidad Del Valle De Mexico, Mexico, compared the performance of 8th, 9th, and 10th graders in 1996 without MATH-TEACHER with 1997 when MATH-TEACHER was first used. The results are listed in the following table.

Grade	Success (%) 1996, without MATH-TEACHER	Success (%) 1997, with MATH-TEACHER	Percent growth in Success
8 th	44	82	86
9 th	74	94	27
10 th	59	87	47

How

Constraints affecting the design: None that we are aware of.

Underlying theories and principles:

- A) Instructional model of mastery learning (Bloom 1981, Block 1971, Carroll 1989)
- B) ACT theory of learning and performance (Anderson 1983, 1990, 1993).

A) CORE IDEA OF MASTERY LEARNING: aptitude is the length of time it takes a person to learn, not how "bright" a person is, i.e., everyone can learn given the right circumstances (Carroll 1989). Mastery learning proposes that all children can learn when provided with the appropriate learning conditions in the classroom. *How to instruct for mastery:* 1. Major objectives representing the purposes of the course or unit define mastery of the subject. 2. The substance is divided into relatively small learning units, each with their own objectives and assessment. 3. Learning materials and instructional strategies are identified: teaching, modeling, practice, formative evaluation, reteaching, and reinforcement, etc., and summative evaluation are included. 4. Each unit is preceded by brief diagnostic tests. 5. The results of diagnostic tests are used to provide supplementary instruction to help student(s) overcome problems. Time to learn must be adjusted to fit aptitude. **NO STUDENT IS TO PROCEED TO NEW MATERIAL UNTIL BASIC PREREQUISITE MATERIAL IS MASTERED.** Bloom, Block, and Carroll believe that mastery learning can be handled in a normal classroom.

B) BASIC TENETS OF THE ACT-R THEORY include: The knowledge required to accomplish complex tasks can be broken down into a large set of simple cognitive skills and facts. Cognitive skill becomes more fluent and robust through practice. Performance on one task will transfer to another task to the extent that the two tasks share skills. Knowledge comes in two types: declarative and procedural. **Declarative knowledge** includes facts, images and sounds. **Procedural knowledge** is an understanding of how to do things. All tasks involve a combination of the two types of knowledge. As we learn, we generally start out with declarative knowledge, which becomes proceduralized through practice. Procedural knowledge tends to be more fluent and automatic. Declarative knowledge tends to be more flexible and broadly applicable. Knowledge is highly interconnected.

The ACT-R theory has clear implications for the design of **educational materials**. Materials should focus on helping students build appropriate conceptual representations and on strengthening the underlying cognitive skills required to master the target tasks. Specifically, educational materials should be effective if they: Are based on research that reveals the skills underlying the target domain of knowledge. Present instruction, tasks and experiences that allow students to learn and practice these skills. Provide immediate feedback, so that errors are attributable to specific cognitive skills. Continually assess students' knowledge of each of the skills, in order to focus each student's instruction on the skills that that student needs to learn and avoid spending too much time on skills that the student has already mastered. Build on what students already know, so that new declarative knowledge is well incorporated into students' knowledge networks. Ground procedural activities in conceptual understanding, so that this knowledge can be proceduralized.

Reference [Anderson]

Start of the design and number of versions: MATH-TEACHER PLUS is the only product of Math-Kal, a company that founded in 1992 by a group of mathematics teachers. We can assume that the start of the design of the MATH-TEACHER series is about 1992 and we study the fourth version of the product.

Development status (advanced prototype, beta product, product): MATH-TEACHER PLUS is a final product.

For whom

Intended Audience: Speaking about a full package of MATH-TEACHER PLUS including Algebra 1, 2, Coordinate Geometry, Trigonometry and Calculus we can say that its broad curriculum is targeted at US Junior High School, High School and First year College. For the Greek Curriculum the target is about the same, from ages 11 to 18 (Greek system consists of 6 years elementary school, ages 6 to 12, 3 years gymnasium, ages 12 to 15 and 3 years Lyceum ages 15 to 18). Algebra 1 and 2 packages focus in Algebra taught at 1st grade gymnasium to 2nd grade Lyceum that is ages 12 to 17.

Distribution

License: MATH-TEACHER PLUS products are available for schools as a site license for a network and stand-alone machines, or lab pack (5 CDs). It is also available on CD for individuals, homeschoolers and other students and teachers.

Price: (http://www.mathkalusa.com/merchant2/merchant.mv?Screen=PLST&Store_Code=MATHKAL)

Each of the five components:	Single CD: \$99
	Complete curriculum (5 CDs): \$395
	Complete curriculum (1 CD): \$300
	Site license: \$795
	Lab-pack: \$375

Artefact

Technical aspects

Operating System (or Web): IBM or compatible PC, Windows 95/98, Windows NT, XP, Windows 2000, 2003. The package can be installed in a network and in stand alone stations.

Technical Requirement:

- Pentium 266 Mhz or higher processor
- 32MB RAM
- 33MB hard disk space
- Graphic card with true color (16 bit or higher)
- Win 98,2000,and XP

Programming Language: multiple

Translations: English, Spanish, Portuguese, French, German, Swedish, Turkish and Hebrew.

[Hickerson Education, Inc. owns the English, French, & Hebrew. Version 4 has not been translated into other languages.]

Interface

Representations:

- what representations

MATH-TEACHER PLUS uses the representations of the typical (formal) algebraic language but there is also the graph representation of several functions and algebraic expressions.

- **nature (usual, new)**

Usual

- **quality (poor, medium, rich)**

Rich

Manipulations:

- * **on what (objects, rules...)**

In MATH-TEACHER PLUS, students manipulate algebraic rules and expressions while solving the exercises presented by the program. They can also experiment with graphs of functions while working in the exploration environment.

MATH-TEACHER PLUS has an algebraic calculator that imports expressions from the answer line in the Tutorial. It returns rational and approximate real values. The calculator includes a button "To answer line" that students use to move the result back to the Tutorial.

- **quality (pleasant, painful):**

Pleasant

Domain

Mathematical content concerning packages Algebra 1, 2 and Word Problems & Probability:

MATH-TEACHER PLUS Algebra-1, 2 and Word Problems & Probability are three different MATH-TEACHER PLUS series and each one consists of two programs: A math program for students and a toolkit program for teachers and administrators. The student program integrates two components: an investigative and reasoning computerized environment, and a tutorial environment, while the Toolkit program integrates three different functions into one program: student management systems, customization utility, and printed test generator.

Tutorials of Algebra 1 cover four areas: Algebraic Expressions - First Steps, Algebraic Expressions – Intermediate, Linear Equations & Inequalities of the First Degree, and Area & Volume.

Tutorials of Algebra 2 cover also four areas: Equations & Inequalities of the Second Degree, Equations & Systems with Two Variables, Quadratic Systems & Parameters, and Sequences.

The tutorials of Word Problems & Probability cover four areas: Problems on Percentages & Numbers, Problems on Geometric Shapes & Motion, Probability – Calculations, and Probability - Combinatorics & Distributions.

Sort of exercises

- **formal exercises:**

Most of the exercises are formal algebra exercises that have a clear aim to help students to understand and practice algebraic rules. For example:

- Calculate the value of the expression $-x^2+3$ for $x = -2/3$
- Simplify the expression $-3x-5+7-2x$
- Write a number greater than $x+3$ by 30%
- Factor the expression $-3x^2+9x$
- Expand $(2x-3)^2$, $(x+7)(x-7)$, $(-a+2)^3$
- Factor x^2-9 , x^2-4x+3 , $8x^3-27$
- Translate the sentence into an expression: If we divide the natural number **m** by **5**, we get a quotient **k** and remainder **b**
- Add the expressions $(2x)/(x-1)-3/(x^2-1)$

- **word problems:**

There is a special unit that presents word problems that can be solved directly by using formulas, or by solving a linear equation with one variable. Students are given percentage problems related to daily life. It also presents problems related to numbers where the relationship between the numbers has to be translated into a linear equation. For example:

- In May the price of tomatoes was **\$50/ton**. Six months later it was marked up by **\$15.81**. What was the percentage of increase in the price?
- The sum of two numbers is 15. When one of them is divided by **2** the result is the other. Find the numbers.

Types of activities:

There are four types of activities in MATH-TEACHER PLUS: lecture, examples solved by the ILE, exercises to be solved by the student, and exploration of functions.

When MATH-TEACHER PLUS is opened, the student can enter the "Exploration Environment" or the "Tutorial Program." Within the Tutorial, the curriculum is organized in units, subjects and levels. Students can follow the program in the order given, or teachers can set up a syllabus. There are 20 units in the complete curriculum, each covering from two to five subjects. When the user clicks on a unit, the next screen lists the subjects for that unit and gives him/her the option of seeing the Lesson, the Interactive Tutorial, or taking an Online Test.

Lecture: Math concepts are taught in the Lesson. The left side of the screen gives the text of the lesson.

Examples solved by the ILE; The right side of the screen gives an example, including the answer, for the type of problem taught in that lesson. An "Example" button allows the student to click and see an infinite number of sample problems from simple to complex.

Exercises to be solved by the student: The student goes back to the unit screen and clicks "Interactive Tutorial" to practice solving problems for a given subject. In a window at the right side of the screen, the title of the chosen subject appears together with sets of exercises organized by the level (up to 5 levels), each set being illustrated by an example. For each level, a minimal number of problems to solve and a minimal grade to obtain are set up in order to be able to move to the next activity. The teacher can change these parameters to adapt them to the students. MATH-TEACHER PLUS is artificially intelligent so the student can solve a problem by typing any number of mathematically balanced statements. Students can solve a problem in two or three steps or up to 20 steps, depending on their ability. When the Tutorial is complete, the program tells the student his grade (*). If the student is ready, he can then take the Online Test. Once the work has been completed, the student or teacher can click on the Report button to view information about the student's progress.

(*) No mistake in the solution process allows the student to obtain 100 points, one mistake 75 points and 2 mistakes 50 points. With 3 and more mistakes, there are no points given. If the solution of a problem requires several steps, each step allows gaining up to 100 points. The grade is the average amount of points on all problems solved so far.

The **Exploration Environment** allows students to see functions and expressions. The student has a variety of math functions and tools to work with, to see everything from the reflection of a line to determining the derivative. Students don't draw graphs. Instead they see how changes in expressions cause changes in the graph.

Feedback

Immediate

Feedback in tutorials is immediate: the program provides feedback on each step of the solution. After the user types the answer and presses <Enter>, the program provides feedback. One of several responses would be “Correct, continue...” if the content of the answer line is a correct answer but still not in the final form. Another feedback is “Incorrect...” if the answer is wrong.

For example, if the problem is to solve the equation:

$x - 3 = 5x - 4$ typing $x - 5x = -4 + 3$ would yield “Correct, continue” as a response, but typing $x + 5x = -4 + 3$ would yield “Incorrect...”. The program decides when the problem is solved by providing the feedback “Correct!” and activating the button that allows moving on.

Each answer line is colored green or red. A correct answer, even if it is not in its final form, is colored green. An incorrect answer is colored red. Each time an answer line is colored (red or green) a new answer line appears in the worksheet area. There is no way to change the previous answer lines.

Using the up-arrow (●), the student can easily obtain the original expression or equation or the previous answer line so that it is entered into the next answer line. The expression can be edited by using the backspace or arrows to move through the expression.

The learner can make up to three mistakes during the solution process. The solution process may require several steps and each student may need a different number of steps. The learner can type many correct steps, however, each mistake is counted and a total of up to three is allowed per question. If he/she makes three mistakes, all steps of the answer will automatically be displayed.

In the process of solving an equation, the student can type a set of equivalent equations until he reaches the final solution. Once he/she has completed, successfully or not, solving this problem, press the “Next Problem” button located on the left bottom corner, to continue to the next problem. The level of difficulty for the next problem is partially random and partially based on student performance. This ensures that the student will solve a variety of problems prior to proceeding to the next level.

If he/she makes two or three mistakes, he/she will be presented with a similar problem, giving him/her another chance to practice the same material. Otherwise, the student will be presented with a slightly more difficult problem. An infinite number of problems can be generated during the practice phase. Each problem is different from previous ones. The algorithm that produces problems generates a different problem each time.

Syntax error is not counted. The program alerts when a syntax error is made by a special message. For example, typing: $x - 5x = -4$ would yield a “syntax error” because the expression is illegal. If the student makes a syntax error he/she can re-edit the line and fix the error. Any answer that is mathematically correct is accepted and colored green. Moreover, the program suggests an interpretation of the algebraic expression typed by the student when there is a possible ambiguity or when the expression involves fractions. For example, typing “ $4 - 7 * (-2/3)$ ” yields the feedback “The expression is interpreted as $4 - 7 * (-\frac{2}{3})$. You can use parentheses to change it if needed”.

Nature:

In addition to the immediate feedback described above, there is mathematical help if needed in order to solve a problem. **Lesson, Guidance and Hint** buttons provide help on how to solve the problem. Click the Lesson button to see a list of the sections for that lesson. Users can choose the relevant section, read the theory, follow the examples and deduct how to solve the current problem. More specific help is the Guidance which provides a general idea how to start solving the problem. This is a static hint that does not change during the solving process.

For example, in the task “Find the numerical value of $\frac{3x}{8x-9}$ for $x = 3$, the guidance message that is displayed on request is: “This is a suggestion how to begin ... Replace x by 3 in the expression.” The most focused help is the Hint that refers to the last answer line and provides a dynamic help on what was wrong if the line was incorrect, or a tip on how to continue from this point if the line was correct.

A **Formula** button is available whenever there are a number of formulas to be considered, e.g., trigonometric identities. When used, it opens a window with a list of relevant formulas.

Diagnosis?

Interactivity

Microworld: No

Hypermedia: Yes

MATH-TEACHER PLUS is a hypermedia that provides a vehicle for intuitive, non-linear access to information and program navigation that more realistically resembles intelligent behaviour.

Answer

Multiple choice questions: No, MATH-TEACHER PLUS does not offer multiple-choice problems; the program allows problems to be solved in a way similar to the way we would do them on paper.

Final open answer: No

Step by step answer: Yes

Features of the tutorial section:

- 1) The students do not have to provide the final answer at the beginning, they can write their answers in steps.
- 2) Each step of the solution is evaluated and feedback whether it is a correct step, or not, is provided.
- 3) During the solution process there are GUIDANCE (an idea how to start solving) and HINTS (dynamic feedback) that provide help to continue with the solving process.
- 4) Students can type their answers in a flexible way. For example, $2x$, $x+x$, $8x/4$ are all equivalent expressions.
- 5) Any correct path that leads to final answer is accepted.

No answer: MATH-TEACHER PLUS does provide printed detailed solutions for printed homework or tests. It is an option when printing problems.

Autonomy

Degree of student's autonomy: There is a limited autonomy for a student who works on his own (e.g. at home), he/she can choose the topic he/she wants to study and decide to take a test when he /she feels ready for it. There is little autonomy for the student if he/she works in a classroom where the teacher chooses the syllabus and sets the parameters that control the behaviour of the program.

Self-evaluation: The program evaluates students' progress. When the Tutorial is complete, the program tells the student his grade. If the student is ready, he can then take the Online Test. Once work has been completed, the student or teacher can click on the Report button to view information about the student's progress.

Knowledge of the task

The ILE knows what is the student's task: Yes

The ILE can provide hint: Yes, there is a Hint button that provides specific help for the last line of a solving procedure (see Feedback section).

The ILE can provide the solution: Yes, when the student makes three mistakes in the solving process the solution is listed and detailed solutions can be printed for printed homework or tests.

The ILE can solve the exercise step by step: In Tutorial programme the student can see many examples solved by the ILE in a step by step mode, but there is no such an option when students solve problems in Interactive Tutorial or in on line tests except as noted in the previous paragraph.

Teacher's role

Parameterization The Toolkit program integrates three different functions into one program: customization utility, student management systems and printed test generator. All three utilities are to be used by teachers.

a) The customization utility

The goal of this utility is to enable teachers to set the parameters controlling the behaviour of the tutorial program for the students of MATH-TEACHER PLUS.

There are a few basic *parameters* that have been pre-determined by Math-Kal but they can be changed by teachers. These parameters apply to each combination of *Unit-Subject-Level*. For each level there are four numbers that control the flow of the program:

- Minimum number of problems the student needs to solve before he/she can progress to the next level.
- Minimum grade the student needs to get in this level before he/she can progress to the next level.
- Activity mode of the calculator in the interactive tutorial (enabled/disabled).
- Number of problems in this level of the test.
- Duration, in minutes, assigned to each problem in this level of the test.
- Activity mode of the calculator in the online test (enabled/disabled).

b) Printed Tests Generator

This utility is a tool that enables teachers to print out lists of problems and answers on each unit and subject. It allows teachers to build and print customized tests and homework for the students. A test can be composed from several units, subjects and levels. The teacher can view the generated list and pick only the problems he/she really wants. The teacher can produce as many lists that have the same level of difficulty, but include different sets of problems.

c) Student Management System (SMS)

This part of the toolkit provides tools for administrating students' records. The management system uses databases, in which all data about the students' activities are recorded. The tool usually is used in two modes:

- An initial mode has to be performed once per each class and can be carried out by a system administrator or teacher. In this mode, class IDs are assigned to each class and student IDs are assigned to students within the class.
- A daily/weekly operating of the utility by teachers, to view students' records, printout reports and statistics, and to backup the database files.

Production of scenarios: No

Production of exercises: Yes

The demo Toolkit will not make customized activities, but it will allow the user to create and save Word documents. It will also save user "MTK TYPE FILES".

The full version allows teachers to create Word documents from the Toolkit. The Word files include experiments/exercises that the students are to complete in the Exploration Environment. Answers can then be exported back into Word for teacher review. Samples Word files are included with each CD.

<i>Validation, usage, experimentation</i>	
<p>Experimentation. I could not find any.</p> <p>Usage analysis? See the results above.</p> <p>Qualitative results?</p> <p>Quantitative results?</p>	
<p>Usage</p> <p>Used in schools and by a great number of homeschoolers.</p> <p>Familiarization with the ILE</p> <p>MATH-TEACHER PLUS is helpful, friendly and easy-to-use software package. MATH-TEACHER PLUS works as a private tutor leading students from theory and exploration to mastery of math concepts.</p>	
<p>Evolution?</p>	

References

- Anderson, J. R. (1983) *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.
- Anderson, J. R. (1990) *The Adaptive Character of Thought*. Hillsdale, NJ: Erlbaum.
- Anderson, J. R. (1993) *Rules of the Mind*. Hillsdale, NJ: Erlbaum.
- Block, J. H. (1971) *Mastery Learning: Theory and Practice*. New York: Holt, Rinehart & Winston.
- Block, J. H., Efthim, H. E., & Burns, R.B. (1989) *Building Effective Mastery Learning Schools*. New York: Longman.
- Bloom, B.S. (1981) *All Our Children Learning*. New York: McGraw-Hill.
- Carroll, J. B. (1963) A model of school learning. *Teachers College Record*, 64(8), 723
- Carroll, J.B. (1989) The Carroll model: A 25 year retrospective and prospective view. *Educational Researcher*, 18(1), 26-31.
- Koedinger K. R. (1998) *Intelligent Cognitive Tutors as Modeling Tool and Instructional Model*. Position Paper for the NCTM Standards 2000 Technology Conference June 5-6, 1998. http://www.carnegielearning.com/web_docs/Intelligent%20Tutors.pdf

TELMA workpackage: ILE study

MathXpert

A study of **Michele Cerulli**, ITD-CNR, Genova, Italy

Design and development

Name of the ILE: MathXpert

Date of the study: June 2006

Who

People/organisation involved in the design

Design and implementation:

*Michael Beeson
Department of Mathematics and Computer Science
San Jose State University
San Jose, California 95192
USA*

Book dialog:

Dave Empey

Splash screen art:

Cindi Shih

Interaction between these participants

Why

Aims of the design

“MathXpert is a computer program designed to help students learn algebra, trigonometry, and one-variable calculus. Its scope extends from elementary algebraic manipulations to the most complicated limits and integrals usually considered in University classes. It is capable of generating complete, step-by-step solutions to mathematical problems, but normally this capability is used to assist the student in developing his or her own step-by-step solution.” (Beeson, [1])

Research questions

How

Constraints affecting the design

According to the author: “Mathematical literacy is much rarer in our society than verbal literacy. Two main reasons for this are

- You have to be very careful in mathematics, because a slight error can throw you completely off track.
- Mathematics is cumulative. You must master each part of the subject before moving on to the next part.” (Beeson, [2])

As a consequence the software has been designed looking at two constraints which are assumed to be characteristics of Mathematics:

1. in mathematics any error can bring the solution of a problem away from the solver
2. Some parts of Mathematics rely on the knowledge of other parts of the subject.

Underlying theories and principles

“MathXpert has been designed from the beginning to satisfy these principles:

- *Correctness.* You cannot make a mathematical mistake.
- *Transparency.* You can see and understand every step, not just the final answer.
- *Appropriate Step Size.* Detailed steps for beginners, powerful short steps in advanced topics.
- *Real Problem Solving.* Not just an electronic page turner, MathXpert can help you solve any problem, including those in your homework assignment.
- *Easy and Accurate Graphing.* The computer can make correct graphs instantly, and change them instantly so you can see how the graph depends on the numbers in the formula. You are never more than one click away from a relevant graph.” (Beeson, [2])

Start of the design and number of versions

Analysing version 3.02 presumably the latest, copyright 2004.

The author declared to have started to work on MathXpert in 1985. He estimates it required seven man-years of effort over the 12-year period 1985-1997.

Development status (advanced prototype, beta product, product)

Product

For whom

Intended Audience

Students (both high or low level students) and teachers. It is meant to be used either at home or in class, and it also intends to substitute blackboards and homework, but not teachers and books.

Distribution

They sell different packages, divided in terms of license or addressed mathematical content.

License

- 14 days trial version (free)
- Stand alone license (prices below)
- Lab licenses which are good for a single-room lab of up to 35 computers. (prices below)

Price

MathXpert Calculus Assistant \$89.95

MathXpert Precalculus Assistant \$69.95

MathXpert Algebra Assistant \$49.95

MathXpert Calculus Lab License \$1750.00

MathXpert Precalculus Lab License \$1350.00

MathXpert Algebra Lab License \$750.00

Artefact

Technical aspects

Operating System (or Web)

MathXpert runs on any version of Windows.

It can be run on a Mac using Virtual PC

It can be run on Linux using WINE.

Programming Language

C

Translations

English

French

German

Interface

Representations

* what rep.

Represents mathematical objects involved in algebra, pre-calculus and calculus. Here we focus only on algebra. It represents:

- algebraic expressions of any kind

- equations and inequalities
- systems of equations and inequalities
- solutions of (systems of) equations and (inequalities)
- all such objects are represented both, as algebraic symbols and as graphics on the Cartesian plane.

*** nature (usual, new)**

The representations are **usual** in terms of how they look like.

However the representations in algebraic symbols are **new** in terms of interactivity as they provide users with context sensitive menus to help choosing what action to perform upon each object

The graphic representations are **usual**.

*** quality (poor, medium, rich)**

The quality of the algebraic representations is **rich** especially in terms of interactivity.

The quality of the graphic representations is **medium**.

Manipulations

*** on what (objects, rules...)**

Algebraic expressions, systems, (systems of) equation, (systems of) inequalities, graphics

*** quality (pleasant, painful)**

Pleasant

Domain

Mathematical content

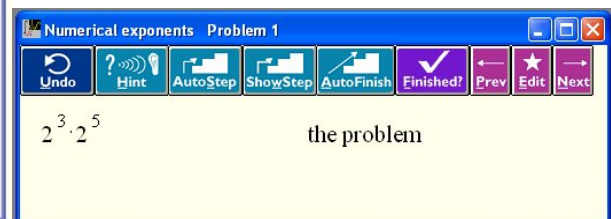
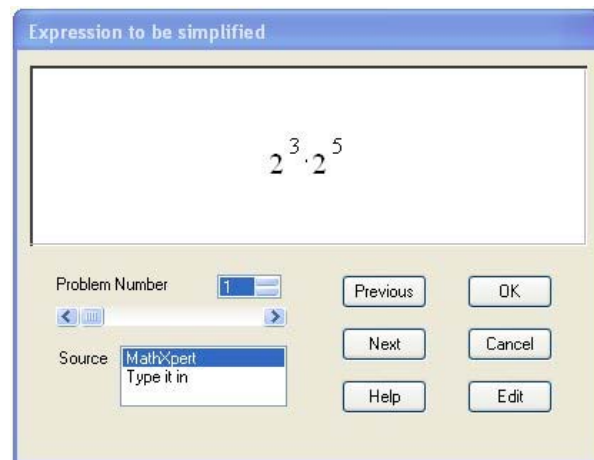
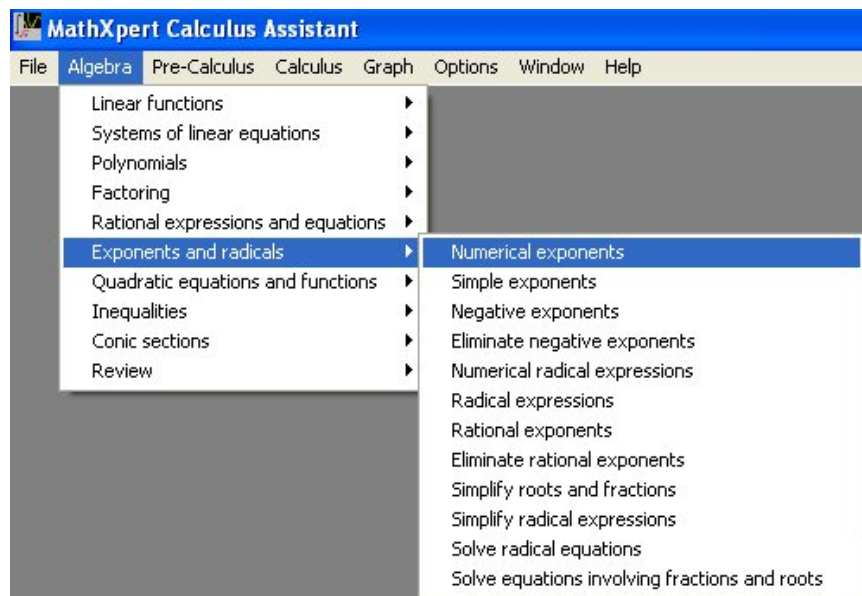
Algebra + Pre Calculus + Calculus

There is a basic assumption that the addressed contents can be organized in levels, so that to learn a given content one needs to know all the previous contents. Following this assumptions the contents are presented in a sequential menu. Moreover within a sub-menu of a given content, problems are presented organized with increasing difficulty.

Sort of exercises

* formal exercises

The exercises are named “problems” and are **formal**, in the sense that they consist of a mathematical object, such as for instance and algebraic expression or equation. The “problems” are organized in menus that provide tasks associated to the objects, and such tasks also appear in the window for editing/choosing problems, but do not appear constantly in the window where the problem is being solved. Examples are shown in the pictures below.



* word problems

Types of activities

The types of activity recall the standard types of activity usually addressed in school practice, such as for instance simplifying or factoring expressions and solving equations.

It is possible to accomplish also tasks concerning manipulation of graphs.

Lecture

There are textual context based hints for solving a given problem: the Hint command opens a popup suggesting what actions to perform to move a single step toward the solution of the problem.

The command ShowStep gives a hint with an animation showing the actions to be performed (instead of a textual popup) to move a single step toward the solution.

No hint is provided in the graphical window.

Examples solved by the ILE

Any problem can be automatically solved by the ILE using the command AutoFinish.

It is also possible to have the ILE bringing forward only one step of solution at time, with the AutoStep command.

No example is provided in the graphical window.

Exercises to be solved by the student

Any problem is supposed to be solved by the student, with the help of the software, as described above (Lecture, Examples solved by the ILE).

Feedback

Immediate/delayed

The feedback received during manipulations is **Immediate**, either in the graphical or symbolic environments.

However, there is particular feedback which is provided to the user on demand by the *Finished* command of the symbolic environment: such a feedback tells the user if the problem can be considered solved or not.

Nature

Most of the feedback received during manipulations is a basic standard feedback needed to make manipulations feasible (ex. selected expression is highlighted).

However there is a particular contextual feedback obtained whenever an expression (or term) is selected: a popup shows a menu where the user can choose the possible action to be performed on the selected expression (or term). Only actions that are compatible with the expression are shown, and the list of actions depends also on the level of difficulty of the exercise (according to how the exercises are classified in the menu of the ILE).

The feedback of the *Finished* command is textual and only tells if the problem is finished or not by stating "Not yet solved" or "That's the answer". Such feedback is on demand, appears only when the users requires it.

Diagnosis

There is no diagnosis of the user.

Interactivity

Microworld

It is a microworld where it is possible to build and manipulate mathematical objects such as functions, graphs, polynomials, algebraic expressions, equations, inequalities, systems, etc..

It is subdivided into two main microworlds, the graphical environment and the symbolic environment (names given by the reviewer) which appear in windows, and it is possible to open several instances of them and address several problems on different windows.

Problems (included functions to be graphed) are edited or inserted into the system through a dedicated input/edit window (which we'll call "problem editor window")

The main kinds of typologies of interaction are: selection and right click, selection and drag (only in the graphical environments), selection and click on buttons, filling of forms, bar slides, etc.

Hypermedia

Only the starting menu for choosing what to do is a sort of hypertext whose leafs open *problem editor windows*.

Answer

Multiple choice question

Final open answer: the user gives his answer when clicks on the Finished command and the answer consists of the last algebraic expression produced, the last line of his work.

Step by step answer

No answer

Autonomy

Degree of student's autonomy

There is a high degree of autonomy to manipulate objects within the microworld using the provided commands. The available commands depend on the level of the addressed contents.

However, a student is free to perform available operations, but all of them are mathematically correct and produce correct expression (perhaps not optimal for solving the problem, but correct): the user is free to choose a wrong or not optimal strategy but is not free to make errors.

Self-evaluation

There is no evaluation at all, if we exclude the feedback provided by the *Finish* command.

Knowledge of the task

The ILE knows what is the student's task

Yes (otherwise the Finish command wouldn't make sense)

The ILE can provide hint

Yes

The ILE can provide the solution

Yes

The ILE can solve the exercise step by step

Yes

Teacher's role

Parameterization

Yes: the teacher (as any user) can change the exercises provided by the ILE changing the parameters

Production of scenarios

No

Production of exercises

Yes: the teacher (as any user) can insert new exercises

Validation, usage, experimentation

Experimentation

Usage analysis

Qualitative results

Quantitative results

Usage

Familiarization with the ILE

Evolution

Refrencies :

- [1] Beeson, M.: *MathXpert: software to help you learn mathematics.*
<http://www.helpwithmath.com/about.php?include=../English-ste.html>
- [2] Beeson, M.: *MathXpert: software to help you learn mathematics.* Author's note.
<http://www.helpwithmath.com/about.php?include=essay.html>.

TELMA workpackage: ILE study

MathXpert

A study of **Mirko Maracci**, University of Siena

<i>Design and development</i>	
Name of the ILE: MathXpert	
Date of the study: August 2006	
Who	
People/organisation involved in the design	
<i>Design and implementation:</i>	
Michael Beeson, Dave Empey, Cindi Shih	
Interaction between these participants	
Why	
Aims of the design	
Research questions	
How	
Constraints affecting the design	
Underlying theories and principles	
Start of the design and number of versions	
MathXpert3.02.	
Development status (advanced prototype, beta product, product)	
Product	
For whom	
Intended Audience	
Distribution	
Different packages (Algebra, Pre-Calculus and Calculus) and different kinds of licence (Trial, Assistant or Lab Licence) are available.	

License

- 14 days trial version (free)
- Stand alone license (prices below)
- Lab licenses which are good for a single-room lab of up to 35 computers. (prices below)

Price

MathXpert Calculus Assistant \$89.95

MathXpert Precalculus Assistant \$69.95

MathXpert Algebra Assistant \$49.95

MathXpert Calculus Lab License \$1750.00

MathXpert Precalculus Lab License \$1350.00

MathXpert Algebra Lab License \$750.00

For this ILE study the authors provided us a Free MathXpert Calculus Assistant Licence

In this study we only refer to **MathXpert Algebra Assistant**

Artefact

Technical aspects

Operating System (or Web)

MathXpert runs on any version of Windows.

Programming Language

Translations

English

French

German

Interface

Representations

*** what rep.**

Several environments are available:

1. The general menus of MathXpert including those for choosing the specific section of problems which one wants to work on (e.g.: system of linear function to be solved by elimination)
2. The problem editor: after having chosen the specific section of problems, users can work on the problems proposed by the ILE or they can edit problems of that specific section: e.g. if one choose the “systems of linear functions” section, then (s)he is expected to edit a system of linear functions, otherwise the ILE provides with error messages or modify the edited problems.
3. The calculation window, that is the workspace where the chosen problem can be worked on.

The workspace is structured into 2 columns: to the left the algebraic objects are represented: numerical and algebraic expressions, equations and inequalities and systems of equations. To the right the trace is recorded of the commands used to transform the algebraic objects, for instance:

$(x+1)(y+2)z$	the problem
$(x+1)(yz+2z)$	$a(b+c)=ab+ac$
$(yz+2z)(x+1)$	$ab=ba$

As one can see in the left column the transformed sub-expressions are coloured.

4. The Graph space where (2D cartesian) graphical representations are provided of:

- Equations (inequalities) with one unknown $a(x)=b(x)$. The graphs of the functions $y=a(x)$ and $y=b(x)$ are displayed.
- Systems of linear equations (inequalities) with two unknowns $a(y)=b(x)$, $c(y)=d(x)$. The graphs of the linear functions $a(y)=b(x)$ and $c(y)=d(x)$ are displayed.
- Polynomials with one variable $p(x)$. The graph of the function $y=p(x)$ and $y=b(x)$ is displayed.

Possible more variables or unknown are treated as parameters which initial (modifiable) value is 1, e.g.: the initial graphical representation associated to the polynomial $(x+1)(t+2)z$ is the graph of the function $y=(x+1)(1+2)1$.

The Graph space is not always available: for instance it is not available for problems consisting in solving algebraic expression.

Moreover each environment contains their own menus, buttons and so on which represent the actions that users can perform in that environment.

* nature (usual, new)

In the calculation window algebraic objects are represented in the usual 2 dimensional form. The only exception is the representation of a system of equations (inequalities) which is represented by means of a couple of square parentheses containing the equations (inequalities). This representation is quite unusual in the Italian schools.

In the problem editor algebraic objects are represented in the 1 dimensional form.

In the graph space both graphical and algebraic 2 dimensional representations are provided.

Command buttons are represented by means of sentences and icons.

* quality (poor, medium, rich)

Manipulations

* on what (objects, rules...)

Numerical and algebraic expressions, equations (inequalities), systems of equations (inequalities), and graphs can be manipulated.

In the calculation window only algebraic signifiers can be manipulated. In order to do that, users have to:

1. select the portion of signifier they want to (users select a rectangular area and the ILE

automatically selects the largest (well-formed) expression, equation... contained in the selected area)

2. select the desired manipulation out from a context sensitive menu which appears as a consequence of the selection: different menus may appear as results of different selections.
3. No free manipulation is possible

In the Graph space, usual manipulations can be performed: changing range, moving axes, re-centering, scaling, zooming...

In the problem editor, users can freely manipulate 1 dimensional algebraic signifiers.

*** quality (pleasant, painful)**

Pleasant

Domain

Mathematical content

The task domains of MathXpert are Algebra, Pre Calculus and Calculus. Here we focus on the first. MathXpert Algebra problems are formal exercises such as: solving (linear or quadratic) equations, factoring polynomials, simplifying radical equations, and so on which also belong to the traditional Italian school practice.

Sort of exercises

*** formal exercises**

The ILE provides formal exercises organized in different sections. Sections differ from each other because they refer to either different algebraic objects (e.g.: linear functions, polynomials, systems of linear equation) or different kinds of tasks, solving algorithms... (e.g.: solve systems of equations by substitution, solve systems of equations by elimination, Cramer's rule).

*** word problems**

Types of activities

Lecture

Examples solved by the ILE

There are not any examples of how to solve a problem within MathXpert, but users can make the ILE solve a problem automatically by means of the AutoFinish command included in the calculation window menu. As a result the whole solution appears in the calculation window menu.

Alternatively one can make MathXpert perform one step of solution by means of the AutoStep command.

Exercises to be solved by the student

All the problems of MathXpert are to be solved by users.

Feedback

Immediate/delayed

All the feedbacks provided by the ILE are immediate, though some of them are 'automatic' and others are 'on demand'.

Nature

The following feedbacks are available within the calculation windows:

- Selected expressions are highlighted. The ILE highlights the largest algebraic expression, equation... in the rectangular area selected by users.
- A context sensitive menu opens as a result of the selection of an algebraic object: the menu shows only the possible (mathematically correct) actions that can be performed on the selected object. Different menus may open when different objects are selected.
- The expression, equation... transformed is displayed together with the explicit reference to the performed action.
- The AutoFinish command shows one solution of the problem.
- The AutoStep command show one step toward the solution of the problem.
- The Hint command opens a popup suggesting what actions can be performed to move a single step toward the solution of the problem.
- The ShowStep command gives a hint with an animation showing the actions to be performed (instead of a textual popup) to move a single step toward the solution.
- The 'Finished?' command checks whether the last line of the whole solution is the expected conclusive step: two different possible feedbacks "Not yet solved" or "That's the answer".

We remark that also the AutoFinish, the AutoStep, the Hint and the ShowStep commands provide context based feedbacks, that is they may show different solutions, further steps, and hints according to the different progresses to the solution made by the user.

Diagnosis

There is no diagnosis of the user.

Interactivity

Microworld

X

Hypermedia

Answer

Multiple choice question

Final open answer: users are expected to provide a final answer in a specific form. For instance.: $2x=1$ or $\frac{1}{2}=x$ are not accessible as final solution of an equation. The expected corrected answer is $x=0,5$ or $x=1/2$

Step by step answer

No answer

Autonomy

Degree of student's autonomy

At each step users have a large number of expression to select and actions to perform, in that sense they have a high degree of autonomy. From the other point of view, their degree of autonomy is in sense limited by the fact that they can only perform mathematically correct actions.

Self-evaluation

No

Knowledge of the task

The ILE knows what is the student's task Yes

The ILE can provide hint Yes

The ILE can provide the solution Yes

The ILE can solve the exercise step by step Yes

Teacher's role

Parameterization

Yes: the teacher (as any user) can change the exercises provided by the ILE changing the parameters

Production of scenarios

No

Production of exercises

Yes: the teacher (as any user) can insert new exercises

Validation, usage, experimentation

Experimentation

Usage analysis

Qualitative results

Quantitative results

Usage

Familiarization with the ILE

The ILE is user-friendly: a short time is needed for familiarizing.

Evolution

Refrencies:

- [1] Beeson, M.: *MathXpert: software to help you learn mathematics.*
<http://www.helpwithmath.com/about.php?include=../English-ste.html>
- [2] Beeson, M.: *MathXpert: software to help you learn mathematics.* Author's note.
<http://www.helpwithmath.com/about.php?include=essay.html>.

TELMA workpackage: ILE study

Ms. Lindquist

A study of **Mirko Maracci**, University of Siena

<i>Design and development</i>	
Name of the ILE: Date of the study: MsLindquist	
Who People/organisation involved in the design <ul style="list-style-type: none"> • Neil Heffernan, Kenneth Koedinger, Cristina Lindquist. Interaction between these participants Neil Heffermann is the ILE's author, he collaborates with Kenneth Koedinger (his Ph.D Adviser) and with Cristina Lindquist (the ILE is named after her). Lindquist was the teacher who has been video taped in class in order to build a tutoring system which was able to behaviour human-like.	
Why Aims of the design MsLindquist is an Intelligent Tutoring System for tutoring students in writing expressions for algebra word problems. The main educational aim is to individually tutor pupils facing the task of writing symbolic expressions for algebra word problems. MsLindquist was not conceived to incorporate and implement tutoring strategies different from those of an "expert human tutor", on the contrary it is an attempt to design an artificial tutor acting as a human one. Obviously, differently from humans MsLindquist can tutor a whole class individually: it is meant to support (or perhaps substitute) teachers in individually tutoring all their pupils. Thus Ms Lindquist is meant to be human-like in carrying on a running conversation, completed with probing questions, worked examples, positive and negative feedback, follow-up questions in embedded sub-dialogs, and requests for explanation as to why something is correct.	
Research questions As far as we could see, no research question is explicitly stated. Anyway from the ILE related documentation (including papers), the concerns emerge of evaluating the effectiveness of the ILE design with respect to the stated educational aims, as well as of comparing such effectiveness with that of other tutoring systems (namely Computer Aided Instruction System and previous Model-tracing Intelligent Tutor Systems). In order to act effectively as a tutor for pupils, the system incorporates a tutorial model with a	

model of student thinking. The latter is used to diagnose student responses. The tutorial model uses the results of this diagnosis to plan a multi-turn tutorial dialog to deal with the particular errors the student made. The dialogs created include embedded feedback (both positive & negative), sub-dialogs, hints and spliced in conversational cues to ensure coherence.

The ILE addresses the difficulty encountered by pupils in symbolization, which is the task of writing an algebraic expression ‘translating’ a word problem. “We found that symbolization is difficult because it is the articulation in the “foreign” language of “algebra”(Heffermann, 2002).

How

Constraints affecting the design

Two main constraints seem to have affected the design process:

1. The design of the ILE was part of the PhD thesis project of the author.
2. The (self posed) aim of making the pupil-tutor interaction as more human-like as possible.

The research was funded by both the [National Science Foundation](#), through a grant to [CIRCLE – Center for Interdisciplinary Research in Learning Environment](#) – , as well as by the [Spencer Foundation](#)

Underlying theories and principles

From the authors’ point of view, the study of the literature related to artificial and human tutoring and the analysis of previous ITSs (Intelligent Tutor Systems) make the need of a new generation of ITSs emerge. Such ITSs are meant to engage in a dialog with students – coherent with the notion of “**knowledge construction dialogs**” (VanLehn) – using multiple strategies to allow students to construct their own knowledge of the domain. MsLindquist is considered by the author as a step towards this new generation tutoring system.

The system expand **the model tracing paradigm**: in fact it is designed so as to incorporate both a model of student thinking – used to diagnose students’ responses – and a tutorial model which starting from the results of the diagnosis is able to plan a multi-turn tutorial dialog to deal with the particular errors the student made. The dialogs created include embedded feedback (both positive & negative), sub-dialogs, hints and spliced in conversational cues to ensure coherence.

Coherently with the **ACT theory of skill knowledge** (Anderson 1993), students’ problem solving skills is modelled as a set of independent rules. The common errors are modelled as “buggy productions”: wrong operator, wrong order of arguments, missing parentheses... The cognitive student model is based on a study aiming at highlighting students’ difficulties in the specific domain and determining their possible errors. The cognitive student model reports its diagnosis to the tutorial model which then chooses the tutoring strategies.

As for the tutorial model, meeting criticism to traditional ITSs, MsLindquist model is meant to be both **tactical** – i.e. able to recognize and driven by students’ local errors – and **strategic** – i.e. able to plan multi-step sequences of tutorial interaction.

More in detail, the tutorial model encodes **pedagogical content knowledge** (Shulman 1986) of an experienced human tutor, in the form of different tutorial strategies. Such strategies were developed by observing an expert human tutor.

Five different tutoring strategies have been implemented:

1. Concrete articulation strategy, which consists in asking pupils: to solve a specific numerical instance of the problem, to explain their answer and then to generalize it. It is based on the works of Gluck (1999) and Koedinger & Anderson (1988).

2. Subgoal reification strategy, inspired by the works of Corbett&Anderson (1995), consists in first asking pupils to state their sub-goals in English and in using symbolic expressions afterwards. As for communication, MsLindquist displays pull down menus for composing sentences.
3. Explanation strategy, inspired by the works of Corbett, Sweller, Simon... consists in presenting pupils with the answer and then asking them to explain in English the portions of the answer itself. Communication is made possible by means of pull down menus
4. Decomposition and Substitution strategy, used for multi-operators problems, consists in: decomposing the original problem in simpler ones (with less operators), introducing new variables for re-composing the problem, scaffolding pupils to substitute auxiliary variables with the appropriate expressions.
5. Giving the answer and moving on to a new problem.

Moreover MsLindquist can use embedded multi-step dialogs to respond to specific situations such as missing parentheses, omissions, argument ordering errors.

Start of the design and number of versions

Current version: MsLindquist2.0

Development status (advanced prototype, beta product, product)

Except for the fact that Ms Lindquist is continually updated, it may be considered a finished product.

For whom

Intended Audience

The ILE has been designed to be used by middle school (6-8th grade) and high school (9 or 10th grade) students.

Distribution

License

MsLindquist is web delivered.

No User Licence (EULA, GFDL...) is specified.

The software is patented: U.S. Patent No. 6,634,887

Price

Free (web-delivered)

Artefact

Technical aspects

Operating System (or Web)

The operating systems supported by Ms Lindquist are Windows and Macintosh. A web-page version is also available.

Programming Language

Java Programming Language

Translations

Ms Lindquist is only available in English.

Interface

Representations

*** what rep.**

The worksheet is divided into three environments named 'Problem', 'History', 'Student input'.

The Problem environment makes available the word problem to be "translated" in algebraic form.

The 'Student input' environment is the channel by which pupils can interact with the system. This environment shapes itself according to the kind of interaction (and consequently to the kind of pupils' input) which establishes to manage.

If pupils are expected to produce an expression, the environment is constituted by a blank in which they can freely digit seemingly with no length restrictions (I tried with 100 digits long expression) – pupils are expected to insert expressions but could also insert words or sentences, in this case however the system is not able to interpret the input.

If pupils are expected to produce sentences (for instance when the tutor follows the explanation strategy) the "Student input" environment is constituted by one or three blanks to be filled by means of pull down menus containing phrases and operators to be selected by pupils.

The 'History environment' contains the virtual dialogue between students and Tutor: the system displays in this space both students' inputs and the consequent Tutor's feedbacks, The whole interaction is shown until the problem is completed.

Unfortunately the history environment does not scroll-down automatically, thus when the interactions is long pupils' last inputs and tutor's feedback are not immediately and automatically displayed.

Pop-up windows are sometimes opened by the system to provide information or ask for feedbacks, e. g.:

1. *Dialog Display* windows provide technical information concerning pupils' expected inputs: "Ms. Lindquist is just a computer and does not understand English, so please make sure your answer to is an expression". It may open during the problem solving activity.
2. *Information* windows are opened at the end of each problem to report on pupils' progresses during the past and current tutoring sessions. It appears at the end of each problem.
3. *Select a button* windows ask for pupils' feedbacks directly related to the interaction with the tutoring systems rather than to the tasks proposed. For instance:

“You are making good progress and have finished this lesson. Ms. Linquist tries to learn if what SHE says helps you learn. If someone, like your teacher or your friend, helped you on this lesson please let Ms. Lindquist know so she doesn't think that what she said helped you. There is absolutely no penalty of any kind for either answer. Did just Ms. Lindquist help you?

I DON'T REMEMBER
JUST MS.LINDQUIST HELPED ME
SOMEONE ELSE HELPED ME”

It opens at the end of sections of problems.

Summing up, the ILE represents:

1. written texts: tasks, sub-tasks, tutor's questions, hints, students' explanations
2. numerical and algebraic expressions both produced by pupils and tutor

*** nature (usual, new)**

Texts are displayed in the usual way. On the contrary inserting texts is to be made by means of pull-down menus from which selecting phrases and operators

The way expressions are displayed and have to be digitized is quite natural.

Some remarks are:

The operator for multiplication is '*' and not '×' (5*7 is accepted , 5×7 is not), if one of the factors is a variable the operator can be omitted. The operator for division is the slash key "/" (12/2 is accepted, 12:2 is not).

*** quality (poor, medium, rich)**

Manipulations

• on what (objects, rules...)

Students' possible actions are:

1. Skipping a whole section of problem to move on to a more advanced section.
2. Sending automatically generated reports on their own progress to the system web-database and to their teachers.
3. Inserting natural language sentences by means of pull down menus
4. Inserting algebraic expressions

Expressions (textual or symbolic) can be manipulated just through the usual Copy and Paste commands – as already remarked textual expressions possibly inserted as “free text” are not interpreted by the system.

*** quality (pleasant, painful)**

Given that the type of the exercises to be solved does not require to manipulate an expression, it is difficult to define the quality of this feature.

Domain

Mathematical content

The task domain is “symbolization”, that is the task of writing an algebraic expression for a given word problem.

Sort of exercises

Though the given problems ask to “translate” a word problem in an algebraic expression, during the problem solving activity, according to the tutoring strategies adopted by the system different sub-problems may be posed:

1. expressing a certain numbers by means of others (i.e. to express 5 as $100/20$)
2. explaining in natural language what a given expression denotes (“what does $g+12$ represent?”)
3. answering natural language questions producing natural language answers
4. substituting variables in a given expression with other (more complex) expressions.

* formal exercises

* **word problems** The tasks the students have to accomplish consist in translating a word problem in an algebraic expression.

Types of activities

Lecture

Examples solved by the ILE

Exercises to be solved by the student

MsLindquist is an Intelligent Tutoring System for tutoring students in writing expressions for algebra word problems individually.

Feedback

Immediate/delayed

Feedbacks are immediate: they are given by the Tutor – based on the student model diagnosis – at each time the student send an input to the system, and displayed in the History environment.

Nature

Positive.

Ms. Lindquist uses 5 different strategies to break down problems:

1. Concrete articulation strategy.
2. Subgoal reification strategy.
3. Explanation strategy.
4. Decomposition and Substitution strategy.
5. Giving the student the answer and then moving on to a new problem

A brief explanation of these feedbacks has been given when discussing the theories and principles underlying the ILE design.

Seemingly a single specific tutoring strategy is a-priori assigned to all the problem of each section: for instance, as for (all) the problems of the first section (One Operator Problem) the tutoring strategy adopted by the system is the “Concrete Articulation strategy”. Whereas the choice of the strategy does not depend on the specific errors pupils make; the way this strategy is instanced depends on the pupils’ specific errors as diagnosed by the cognitive student model

Diagnosis

As explained when discussing the theories and principles underlying the ILE design, students’ problem solving skills is modelled as a set of independent rules. In particular students’ common errors are modelled as “buggy productions”: wrong operator, wrong order of arguments, missing parentheses... By means of these productions, the cognitive student model is able to diagnosed pupils’ inputs. The full diagnosis is reported to the tutorial model which then chooses the suitable tutoring strategies.

Interactivity

Microworld

X

Hypermedia

Answer

Multiple choice question

When adopting the ‘Subgoal reification’ or ‘Explanation’ tutoring strategies – described above – MsLindquist asks pupils to compose English sentences: MsLindquist displays pull down menus for composing sentences. These questions are “de facto” multiple choice questions.

Final open answer

Pupils are expected to produce algebraic expressions.

Step by step answer

When adopting the “Decomposition and Substitution” strategy, the tutor decomposes the original problem in simpler ones (with less operators), introducing new variables for re-composing the problem, scaffolding pupils to substitute auxiliary variables with the appropriate expressions.

No answer

Autonomy

Degree of student’s autonomy

As previously said, pupils can perform the following actions:

1. Skipping a whole section of problems to move on to a more advanced section. (Very low degree of autonomy in selecting problems)
2. Inserting natural language sentences by means of pull down menus (Low degree of autonomy in composing English texts)
3. Inserting algebraic expressions (High degree)

Self-evaluation

None

Knowledge of the task

The ILE knows what is the student’s task

The ILE chooses students’ task.

The ILE can provide hint

The ILE chooses the hints to provide to pupils.

The ILE can provide the solution

The ILE can provide the solution, but only if the system itself chooses such tutoring strategies. It is not on pupils to decide whether they can see the solution, it is a prerogative of the system.

The ILE can solve the exercise step by step

No, the ILE can provide with hints step by step.

Teacher's role

MsLindquist is conceived to tutor pupils in a completely autonomous way.

The system decides not only the tutoring strategies, but also the problems to propose, the order according to which problems are given, the starting section and so on

Parameterization

Production of scenarios

Production of exercises

No exercise editor is available, though in principle one who knows the programming language can add more problems (in the on-site version)

Validation, usage, experimentation

Experimentation

Usage analysis

Qualitative results

In 2001 the authors carried out a study aiming at comparing “the instruction Ms Lindquist provides with a traditional computer-aided approach”.

20 students have been involved in the study divided in 2 groups: one group was given a 2 hours session of the instruction provided by MsLindquist, the other group was given a 2 hours session of traditional computer aided instruction.

The only difference according the authors is that:

- **Ms. Lindquist** is able to carry on a running conversation, complete with probing questions, positive and negative feedback, follow-up questions in embedded sub-dialogs, and requests for explanation as to why something is correct.
- **Traditional Computer-Aided Instruction(CAI)** simply told the students the answer if the student got a question wrong. You can think of this condition being analogous to a student doing homework with the answers in the back of the book but no additional help.

Everything else was held the same between the two conditions including the number of problems done.”

“Students were given a 6 item post test after about 2 hours of instruction. Students using Ms. Lindquist did better on the post test (The effect size of .8 was large and we had a p value of .2. We only had 20 students.).”

According the authors “these results show that the dialogs Ms. Lindquist engages students in is effective in promoting student learning.”

Quantitative results

Usage

Familiarization with the ILE

The ILE is user-friendly: a short time is needed for familiarizing.

Evolution

TELMA workpackage: ILE study

Ms. Lindquist

A study of **Michele Cerulli**, ITD-CNR, Genova, Italy

Design and development

Name of the ILE: Ms. Lindquist

Date of the study: August 2006

Who

People/organisation involved in the design

- Neil Heffernan (<http://www.cs.wpi.edu/~nth>)
- Neil Heffernan's Ph.D. Adviser [Kenneth Koedinger](#) who helped create [Cognitive Tutor Algebra](#) which was named one of the five [Exemplary Programs](#) by the U.S. Dept of Education.
- Ms. Cristina Lindquist was the expert human tutor who was video taped. The tutor is named after her.
- Funded by: [The National Science Foundation](#) through a grant to the [CIRCLE](#) center.
- Funded by: [The Spencer Foundation](#)

Interaction between these participants

Why

Aims of the design

To allow overcoming students' problems to solve word problems and in particular to write algebra expressions for algebra word problem. In class a teacher, in a one-on-one relationship with pupils could break these problems into manageable pieces, but it turns out to be impossible to do this for a whole class, for this reason it is needed to develop a computer tutor which can be used by all the pupils of a class.

Research questions

Assuming that the single most important mathematical skill for students who wish to study beyond arithmetic is the ability to take a problem situation (usually stated in words) and formulate a mathematical model (usually an equation), the developers call this skill *symbolization* and address the two following main research questions (see also the dedicated web site http://www.cs.cmu.edu/~neil/research_based.html):

- Why is it so difficult for pupils to translate word problems into algebraic symbols? (see Heffernan et al., 1998, [4])
- How is it possible to make a tutoring system to help pupils learning such a symbolization skill? (see Heffernan et al., 2000, [3])

How

Constraints affecting the design

According to the authors in order to build a good ITS for a domain you need to (see, [3]):

1. Study what makes that domain difficult, including discovering any hidden skills, as well as determining what types of errors students make.
2. Construct a theory of how students solve these problem. (They instantiated that theory in a cognitive model.)
3. Observe experienced human tutors to find out what pedagogical content knowledge they have and then build a tutorial model that, with the help of the theory of domain skills, can capture some of that knowledge.
4. Use your model to answer questions about what makes for effective tutoring.

Moreover they assumed that the tutoring system *should provide to the user with a sense of dialog, similarly to the case of human tutoring.*

Moreover the tutoring system should be able *not only to model-trace the student's actions, but also to be more human-like in carrying on a running conversation*, complete with probing questions, worked examples, positive and negative feedback, follow-up questions in embedded sub-dialogs, and requests for explanation as to why something is correct. In order to build Miss Lindquist the developers have expanded the model-tracing paradigm; not only does she have a model of the student but also has a model of tutorial reasoning (e.g. Clancey, 1982) for our specific domain (to do so, the developers analysed human tutors at work).

Finally, the developers argue that a tutor needs to be tactical in recognizing certain situations, and strategic enough to plan multi-step sequences of tutorial interaction. Therefore, *it is important that Miss Lindquist can understand the student and come up with a strategy to respond with, even in the presence of multiple errors.*

Underlying theories and principles

Concerning the educational problem

In order to explain why symbolization is so difficult the developers propose the “*articulation hypothesis* which suggests that there is a "hidden" skill that is not obvious to most teachers and researchers. The hidden skill is the ability to produce symbolic sentences in the language of algebra. It appears that many students are able to figure out all the conceptual relations in a problem, but are not able to express those relationships in algebra. If we asked students to translate a story written in English into Greek we would not be surprised if many fail because they don't know Greek. But teachers and researchers often fail to realize that algebra too is a language. And a language that students have had relatively little practice in "speaking" By "speaking" we mean producing sentences of symbols, not verbalizing” ([3]).

Concerning the student model (see [3])

The developers used a student model which is a cognitive model of the problem solving knowledge that students are acquiring. It reflects the ACT theory of skill knowledge (Anderson, 1993) in assuming that the problem solving skills can be modeled as a set of independent production rules.

Traditional model-tracing tutors generate their feedback from templates associated with individual production rules. Instead, MsLindquist passes the diagnosis to the tutorial model that reasons between tactical and strategic concerns.

Concerning the tutor model

The “tutorial model is informed by observation of an experienced human tutor. The tutorial model is implemented with 77 production rules that decide what to say and how to say it. Many of these productions manipulate the tutorial agenda which operates usually like a push-down stack. Most of the productions are concerned with implementing the various tutorial strategies but some are concerned with more mundane aspects including maintaining conversational coherence, giving positive feedback, and dealing with issues related to how to deal with multiple student errors or incomprehensible student input. This approach is similar to Freedman’s (2000)” (see [3]).

Two further strategies for fostering translations between multiple representations (see [3])

According to the developers “mathematical communication (Schoenfeld et. al., 1992, Koedinger & Nathan, in press) is increasingly seen as a goal for students, and in particular, translating between algebraic (e.g. “ $m/s+h$ ”), English (e.g. “the total trip time”) and concrete (e.g. “7 hours”) representations.”. The educational problem addressed by the developers concerns not only translating from English to algebraic symbols, but also: 1) translating from the algebraic to the English and 2) within the English representation. To address these two kinds of translations, the authors adopted two strategies inspired by the literature on the field:

Subgoal Reification Strategy: “The subgoal reification strategy is inspired by Corbett & Anderson’s (1995) lisp tutor that showed learning gains when the students we asked to post (or reify) their subgoals before attempting to write lisp code. We have added to our tutor the ability to choose a similar strategy”. The strategy can be schematised as follows: - The student is asked for the name of the subgoal.

- The student is asked to explain in English how to achieve that subgoal.
- The tutor asks for the subgoal in algebra.

The pattern is repeated.

Self-Explanation of a Worked Example: this strategy “is inspired by the literature on worked examples and the self-explanation effect. Sweller (1988) proposed giving students worked out examples. Chi (1996) and Conati et. al. (1997) have both proposed asking students to self-explain. This strategy also gets at the goal of practicing translation from the algebraic representation to the verbal representation which Corbett et. al. (2000) have shown to be effective for learning symbolization”. An example could be that of giving the student the answer of the problem and ask him/her to explain it.

Start of the design and number of versions

No found, however the first papers concerning the student’s developmental model implemented in the system have been published by the authors in 1997, which we could take as start of the design.

Numbers of versions: presumably 1

Development status (advanced prototype, beta product, product)

Product

For whom

Intended Audience

Both students and teachers: students to have themselves tutored, and teachers for having a support in tutoring students. The considered school level is not specified, but it is determined by the fact that the system address the transition from arithmetic to algebra in terms of working on how to write algebraic expressions for word problems.

Distribution

License

Free, Web-Delivered, Patented (U.S. Patent No. 6,634,887)

Price

Free

Artefact

Technical aspects

Operating System (or Web)

Web

Programming Language

The interface is a Java application

Translations

Available only in English

Interface

Representations

* what rep.

Verbal representation of word problems and of algebraic formula

* nature (usual, new)

Usual on one line

* quality (poor, medium, rich)

Poor (but ok for the aims of the system)

Manipulations

* on what (objects, rules...)

no manipulations

* quality (pleasant, painful)

Domain

Mathematical content

Word problems with 1, 2 or 3 operators

Sort of exercises

* formal exercises

* word problems

The user is asked to write formulas for solving word problems, the formula is considered to be the solution of the problem

Types of activities

Lecture

Examples solved by the ILE

Yes

Exercises to be solved by the student

Yes

Feedback

Immediate/delayed

Immediate

Nature

Written text

Diagnosis

Appears only indirectly: on the one hand there is a window summarizing the student's progress in terms of solved problems and tests; on the other hand, the tutor's strategy is adapted to the students behaviour.

Interactivity

Microworld

No (no computational objects to act upon)

Hypermedia

It is basically a text window with input forms and multiple choice menus

Answer

Multiple choice question

One of the strategies of the tutor wants to help the student solving a problem is to provide a possible verbal answer in the form of written text constituted by multiple choice menus: the user builds the phrase choosing from the menu

Final open answer

Yes, the user is asked to write algebraic expressions

Step by step answer

In some cases the problem is split into sub problems, and the user is asked to write algebraic expressions

No answer

Autonomy

Degree of student's autonomy

The students has low autonomy in terms of choosing the exercises (which in each section are chosen by the tutor), but has autonomy in writing answers. However the tutor's support is constantly present.

Self-evaluation

None

Knowledge of the task

The ILE knows what is the student's task

Yes

The ILE can provide hint

Yes

The ILE can provide the solution

Yes

The ILE can solve the exercise step by step

Yes

Teacher's role

No particular role is expected to be played by the teacher.

Parameterization

no

Production of scenarios

no

Production of exercises

No

Validation, usage, experimentation

Experimentation

Usage analysis

Qualitative results

The authors brought forward a comparative study where they compared the instruction Ms Lindquist provides with a traditional computer-aided approach.

The variable that was kept different from the two experimental settings is the following:

- **Ms. Lindquist** is able to carry on a running conversation, complete with probing questions, positive and negative feedback, follow-up questions in embedded sub-dialogs, and requests for explanation as to why something is correct.
- **Traditional Computer-Aided Instruction(CAI)** simply told the students the answer if the student got a question wrong. You can think of this condition being analogous to a student doing homework with the answers in the back of the book but no additional help.

Everything else was held the same between the two conditions including the number of problems done.

Students were given a 6 item post test after about 2 hours of instruction. Students using Ms. Lindquist did better on the post test (The effect size of .8 was large and we had a p value of .2. We only had 20 students.).

More information can be found on the author's PhD thesis (Hefferman, 2001, [5] ,Chapter 5)

Quantitative results

Usage

Familiarization with the ILE

The familiarization happens through an introductory phase where on the one hand the system gets information on the user, and on the other hand lets the user try out the system proposing word problems to be solved. During this introductory phase, the system informs the user on the typologies of strategies adopted by the system to help the student. At the end of this phase the system asks the user if he/she is a real student or he/she is a teacher or parent who just wanted to test the system out. This is important for the system as it learns from the user's behaviour assuming that the user is a student and.

Evolution

Stable

References

- [1] Anderson, J. R. (1993). *Rules of the Mind*. Hillsdale, NJ: Erlbaum.
- [2] Freedman, R. (2000 to appear) Using a reactive planner as the basis for a dialogue agent. In *Proceedings of the Thirteenth Florida Artificial Intelligence Research Symposium (FLAIRS '00)*, Orlando.
- [3] Heffernan, N. T., & Koedinger, K. R. (Presented at the Intelligent Tutoring Systems 2000 Conference Workshop entitled "Intelligent Tutoring Systems Conference's Workshop in Algebra Learning"). *Building a 3rd generation ITS for symbolization: Adding a Tutorial Model with Multiple Tutorial Strategies*. ([postscript .ps -11 pages](#)) ([web page version](#))
- [4] Heffernan, N. T. & Koedinger, K. R. (1998). A developmental model for algebra symbolization: The results of a difficulty factors assessment. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, (pp. 484-489). Hillsdale, NJ: Erlbaum. ([postscript .ps](#)), ([html](#))
- [5] Heffernan, N. T. (2001): "*Intelligent Tutoring Systems have Forgotten the Tutor: Adding a Cognitive Model of Human Tutors*". Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, March 2001. School Of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

6. Answers of the authors

ActiveMath

The authors provided a few complementary elements which have been added to the grid.

AnimalWatch

The authors have not yet answered.

Cognitive Tutor

The authors have not yet answered.

Math-Teacher Plus

The authors made a few corrections in the grid.

MatXpert

The author gave a few elements which were already in the grid. On point will be discussed with the author and the result will be integrated in the work of next year.

Ms. Lindquist

The author mentioned an experiment which was not in the study. This experiment will be taken into account in the work of next year.

7. Conclusion and future work

Studying these ILEs have been very interesting and the result has an interest both for the mathematics education community, and the AI and Education community. The work will be presented in an open workshop called “Interactive Learning Environments (ILEs) for arithmetic and algebra education: the TELMA approach” in Grenoble, by November 14, 2006.

This work will continue in the 4th year of TELMA. The grid will be analysed taking into account its use in these studies in order to explicit some point, add or change some points. A few new ILEs will be studied. The result will be placed on the Website of TELMA.

8. References

J.F. Nicaud, D. Bouhineau, H. Chaachoua, J. Trgalova (2006). Developing Interactive Learning Environments that can be used by all the classes having access to computers. The case of Aplusix for algebra. Proceeding of the 17th ICMI Study 'Digital technologies and mathematics teaching and learning: Rethinking the terrain'. Hanoi, Vietnam, 2006.

P. Tchounikine (2004). Platon-1: Quelques dimensions pour l'analyse de travaux de recherche en conception d'EIAH. <http://www-lium.univ-lemans.fr/~tchou/Platon-1.pdf>